สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การสร้างและควบคุมหุ่นยนต์เลียนแบบมนุษย์ขนาดเล็ก

นายปฐม ฤมิตรัชตานนท์
นายสมมาตร วัฒนไพโรจน์

เลขหมู่ 62850
เลขทะเบียน
วัน,เดือน,ปี 23 ส.ค. 2549
ล. 11b32811
ส.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ROBO-ONE

MR. PATHOM SUMITRACHATANON

MR. SOMMART WATTHANAPAIROJ

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INDUSTRIAL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ การสร้างและควบคุมหุ่นยนต์เลียนแบบมนุษย์ขนาดเล็ก ROBO-ONE

นักศึกษา

นายปฐม สุมิตรัชตานนท์	รหัสประจำตัว	45010431
นายสมมาตร วัฒนไพโรจน์	รหัสประจำตัว	45010801

หลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอุตสาหการ

อาจารย์ผู้ควบคุมปริญญานิพนธ์

(อาจารย์พลชัย โชติปรายนกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การสร้างและควบคุมหุ่นยนต์เลียนแบบมนุษย์ขนาดเล็ก
นักศึกษา	นายปฐม สุมิตรัชตานนท์
	นายสมมาตร วัฒเนไพโรจน์
หลักสูตร	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา	2548
อาจารย์ผู้ควบคุมปริญญานิพนธ์	อาจารย์พลชัย โชติปรายนกุล

บทคัดย่อ

โครงงานปริญญานิพนธ์ฉบับนี้มีวัตถุประสงค์ เพื่อสร้างและควบคุมหุ่นยนต์เลียนแบบมนุษย์ขนาดเล็ก ส่วนประกอบของหุ่นยนต์ตัวนี้โครงสร้างทำด้วยอลูมิเนียมแผ่นมีขนาดความหนา 1.5 มิลลิเมตร โดยหุ่นยนต์ตัวนี้ใช้เซอร์โวมอเตอร์ขนาดเล็กเป็นตัวข้อต่อในจุดที่ต้องการหมุนต่างๆ และใช้โปรแกรมควบคุมเซอร์โวมอเตอร์ด้วยไมโครคอนโทรลเลอร์ตระกูล MCS-51 และมีการพัฒนาระบบไร้สายในการควบคุม ซึ่งแนวทางการสร้างหุ่นยนต์ที่ได้สามารถนำไปประยุกต์ใช้งานในด้านอุตสาหกรรม เช่น ออกแบบโปรแกรมควบคุมแขนกลหรือระบบอัตโนมัติที่ใช้ในงานอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	ROBO-ONE
Student	Mr. Pathom Sumitrachatanon
	Mr. Sommart Watthanapairoj
Degree	Bachelor of Engineering in Industrial Engineering
	King Mongkut's Institute of Technology Ladkrabang
Academic Year	2005
Thesis Advisor	Mr. Pholchai Chotipraynakul

ABTRACT

The objectives of this project are to build a small size of humanoid robot and to program its control. All frames the robot are made of 1.5 mm aluminum plate thickness. The major components are servomotors and a MCS-51 microcontroller which to develop of a wireless system. These project can be an application for industrial usage, for example, a robot arm which is controlled by the program, automation systems in industrial works.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง การสร้างและควบคุมหุ่นยนต์เลียนแบบมนุษย์ขนาดเล็ก สามารถสำเร็จลุล่วงไปด้วยดี กลุ่มผู้พัฒนาขอกราบขอบพระคุณทุกคนที่มีส่วนเกี่ยวข้องส่งผลให้ปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์

อาจารย์พลชัย โชติปรายนกุล อาจารย์ที่ปรึกษาปริญญานิพนธ์ กลุ่มผู้พัฒนาขอกราบขอบพระคุณเป็นอย่างสูง สำหรับการให้โอกาสศึกษาปริญญานิพนธ์ฉบับนี้ รวมไปถึงความรู้ คำแนะนำ ตลอดจนสถานที่ และความช่วยเหลือในด้านต่างๆ ในการจัดทำปริญญานิพนธ์ฉบับนี้

รศ.พรศักดิ์ อรรถวานิช หัวหน้าภาควิชาวิศวกรรมอุตสาหการ กลุ่มผู้พัฒนาขอกราบขอบพระคุณเป็นอย่างสูงสำหรับความเอาใจใส่และทุกสิ่งทุกอย่างตลอดการศึกษาระดับปริญญาตรี ในหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอุตสาหการ

ผศ.ดร.สรรพสิทธิ์ ลิ่มนรรัตน์ กลุ่มผู้พัฒนาขอกราบขอบพระคุณเป็นอย่างสูง สำหรับคำแนะนำ ข้อคิดที่จำเป็นต้องใช้ในการทำงานในอนาคต และความช่วยเหลือในทุกๆ ด้านตลอดการศึกษาระดับปริญญาตรี ในหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอุตสาหการ

ดร.สิทธิพร พิมพ์สกุล กลุ่มผู้พัฒนาขอกราบขอบพระคุณเป็นอย่างสูง สำหรับคำแนะนำ ความรู้ และความช่วยเหลือในทุกๆ ด้าน

อาจารย์เชาวลิต หามนตรี กลุ่มผู้พัฒนาขอกราบขอบพระคุณเป็นอย่างสูง สำหรับความรู้ และความช่วยเหลือในทุกๆ ด้าน

ขอบคุณเพื่อนทุกคนสำหรับความช่วยเหลือจนทำให้ปริญญานิพนธ์สำเร็จลุล่วง และคอยเป็นกำลังที่ดีตลอดมา

นายปฐม สุมิตรัชตานนท์
นายสมมาตร วัฒนไพโรจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ

1.1 ความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตการศึกษาของโครงงาน 1

1.4 ประโยชน์ที่คาดว่าจะได้รับ 1

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 เซอร์โวมอเตอร์ 2

2.1.1 ความหมายของเซอร์โวมอเตอร์ 2

2.1.2 หลักการทำงานของเซอร์โวมอเตอร์ 3

2.2 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 (AT89C51) 6

2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51 5

2.2.2 หน่วยความจำ (Memory) 7

2.2.3 พอร์ตอินพุตและพอร์ตเอาท์พุต 11

2.2.4 การทำงานของรีจิสเตอร์ 11

2.2.5 ชุดคำสั่งของไมโครคอนโทรลเลอร์ตระกูล MCS-51 14

2.3 การเชื่อมต่อแบบไร้สาย (Wireless RF Module) 20

2.4 โปรแกรมแอสเซมบลี 21

บทที่ 3 การออกแบบและวิธีการดำเนินงาน

3.1 ศึกษารวบรวมข้อมูล 22

3.2 ออกแบบและเขียนแบบหุ่นยนต์ 22

3.3 ดำเนินการสร้างหุ่นยนต์ 23

3.4 สร้างแผงวงจรไมโครคอนโทรลเลอร์, วงจรรับ-ส่งสัญญาณระบบไร้สาย 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.5 ออกแบบโปรแกรมสั่งการเซอร์โวมอเตอร์ 25

3.6 ขั้นตอนเขียนโปรแกรมและทดสอบโปรแกรม 27

บทที่ 4 ผลการดำเนินงาน

4.1 ผลการดำเนินงานด้านฮาร์ดแวร์ – การสร้างหุ่นยนต์ ROBO-ONE 28

4.2 ผลการดำเนินงานด้านวงจรไฟฟ้า 28

4.3 การทดสอบ 30

บทที่ 5 สรุปและวิเคราะห์ผลการดำเนินงาน

5.1 รายละเอียดของปริญญานิพนธ์ 31

5.2 วัตถุประสงค์ของโครงงานปริญญานิพนธ์ 31

5.3 ประโยชน์ของโครงงานปริญญานิพนธ์ 31

5.4 ผลการดำเนินงาน 31

5.5 แนวทางในการปรับปรุงพัฒนาโครงงานปริญญานิพนธ์ 31

5.5.1 การปรับปรุงในส่วนของวงจร 31

5.5.2 การปรับปรุงในส่วนของโปรแกรม 32

5.5.3 การปรับปรุงในส่วนของโครงสร้างหุ่นยนต์ 32

หนังสืออ้างอิง 33

ภาคผนวก ก ..... ผก 1

ภาคผนวก ข ..... ผข 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 รีจิสเตอร์ RO - R7 8
ตารางที่ 2.2 รีจิสเตอร์หน้าที่พิเศษ (SFR) 9
ตารางที่ 2.3 ค่ารีจีสเตอร์ต่างๆ เมื่อเริ่มต้นทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 13
ตารางที่ 2.3 ค่ารีจีสเตอร์ต่างๆ เมื่อเริ่มต้นทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51(ต่อ) 14
ตารางที่ 2.4 ค่าตัวแปรที่กำหนดในคำสั่ง 15
ตารางที่ 2.5 กลุ่มคำสั่งทางคณิตศาสตร์ 16
ตารางที่ 2.6 กลุ่มคำสั่งการกระทำลอจิก 17
ตารางที่ 2.7 กลุ่มคำสั่งการเคลื่อนย้ายข้อมูล 18
ตารางที่ 2.8 กลุ่มคำสั่งการจัดการข้อมูลระดับบิต 19
ตารางที่ 2.9 กลุ่มคำสั่งการกระโดด 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 แผนภาพเซอร์โวมอเตอร์ 2
รูปที่ 2.2 แผนภาพแสดงส่วนประกอบของเซอร์โวมอเตอร์ 3
รูปที่ 2.3 แผนภาพแสดงสัญญาณความกว้างพัลส์ 4
รูปที่ 2.4 แผนภาพแสดงค่าความกว้างพัลส์และระยะองศาการหมุนของเซอร์โวมอเตอร์ 4
รูปที่ 2.5 แผนภาพแสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 6
รูปที่ 2.6 แผนภาพแสดงสัญลักษณ์ของไมโครคอนโทรลเลอร์เบอร์ AT89C2051 7
รูปที่ 2.7 แผนภาพแสดงพื้นที่หน่วยความจำภายในของไอซี AT89C2051 8
รูปที่ 2.8 แผนภาพแสดงหน่วยความจำภายในบริเวณที่อ้างถึงแบบบิตได้ 9
รูปที่ 2.9 แผนภาพแสดงการจัดหน่วยความจำและตำแหน่งของรีจิสเตอร์เฉพาะ (Special Function Register) 10
รูปที่ 2.10 แผนภาพแสดงรีจิสเตอร์ PSW (Program Status Word) 11
รูปที่ 3.1 แผนภาพการออกแบบข้อต่อที่เลียนแบบท่อนแขนหรือท่อนขาของมนุษย์ 22
รูปที่ 3.2 แผนภาพแสดงส่วนประกอบต่างๆ ของหุ่นยนต์ 23
รูปที่ 3.3 แผงวงจรไมโครคอนโทรลเลอร์และแผงวงจรรับสัญญาณระบบไร้สาย 24
รูปที่ 3.4 แผงวงจรสั่งสัญญาณระบบไร้สาย 25
รูปที่ 3.5 แผนผังแสดงการออกแบบโครงสร้างโปรแกรมในส่วนของโปรแกรมภาษาแอสเซมบลี 25
รูปที่ 3.6 แผนผังแสดงการออกแบบโปรแกรมในส่วนของระบบไร้สาย 26
รูปที่ 4.1 รูปถ่ายแสดงผลการดำเนินงานด้านฮาร์ดแวร์ 28
รูปที่ 4.2 รูปถ่ายแผงวงจรไมโครคอนโทรลเลอร์และแผงวงจรรับสัญญาณระบบไร้สาย 29
รูปที่ 4.3 รูปถ่ายแผงวงจรส่งสัญญาณระบบไร้สาย 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เทคโนโลยีหุ่นยนต์มีบทบาทสำคัญในอุตสาหกรรมการผลิตมามากกว่า 40 ปี จนทำให้หุ่นยนต์ยุคใหม่มีความฉลาดมากขึ้น เพื่อทำงานที่ซับซ้อนได้ดี ซึ่งมีคำกล่าวว่า เทคโนโลยีหุ่นยนต์ เป็นหนึ่งในห้าเทคโนโลยีสุดยอดของอนาคต ทางกลุ่มผู้จัดทำโครงงานจึงมีความสนใจในการวิจัยและพัฒนาระบบหุ่นยนต์ โดยสามารถนำไปประยุกต์ใช้กับอุตสาหกรรม เพื่อพัฒนาเพิ่มผลผลิตในอุตสาหกรรมไทย ซึ่งกลุ่มผู้จัดทำโครงงานจึงมีความคิดที่จะเริ่มต้นศึกษาเกี่ยวกับหุ่นยนต์ด้วยหุ่นยนต์ขนาดเล็กเลียนแบบท่าทางต่างๆ ของมนุษย์ ที่มีข้อต่อต่างๆ ใกล้เคียงกับมนุษย์

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อสร้างหุ่นยนต์เลียนแบบมนุษย์
2. โปรแกรมควบคุมหุ่นยนต์ให้เคลื่อนไหวคล้ายมนุษย์เพื่อควบคุมเซอร์โวมอเตอร์

1.3 ขอบเขตการศึกษาของโครงงาน

1. หุ่นยนต์ยังต้องมีการเชื่อมโยงและควบคุมผ่านไมโครคอนโทรลเลอร์
2. ใช้ไมโครคอนโทรลเลอร์ในการติดต่อกับคอมพิวเตอร์
3. การพัฒนาในรูปแบบไร้สาย

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถอธิบายการควบคุมหุ่นยนต์ที่สามารถเคลื่อนที่ได้แบบมนุษย์
2. สามารถเขียนโปรแกรมควบคุมหุ่นยนต์ให้เคลื่อนที่ตามที่ต้องการ
3. สามารถใช้เทคโนโลยีทางด้านหุ่นยนต์ เป็นกลไกในการศึกษาและแก้ปัญหาการโปรแกรมหุ่นยนต์

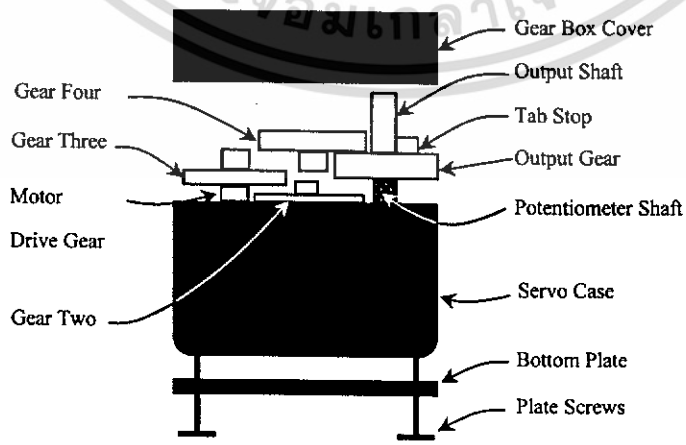
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 เซอร์โวมอเตอร์ (Servo motor)

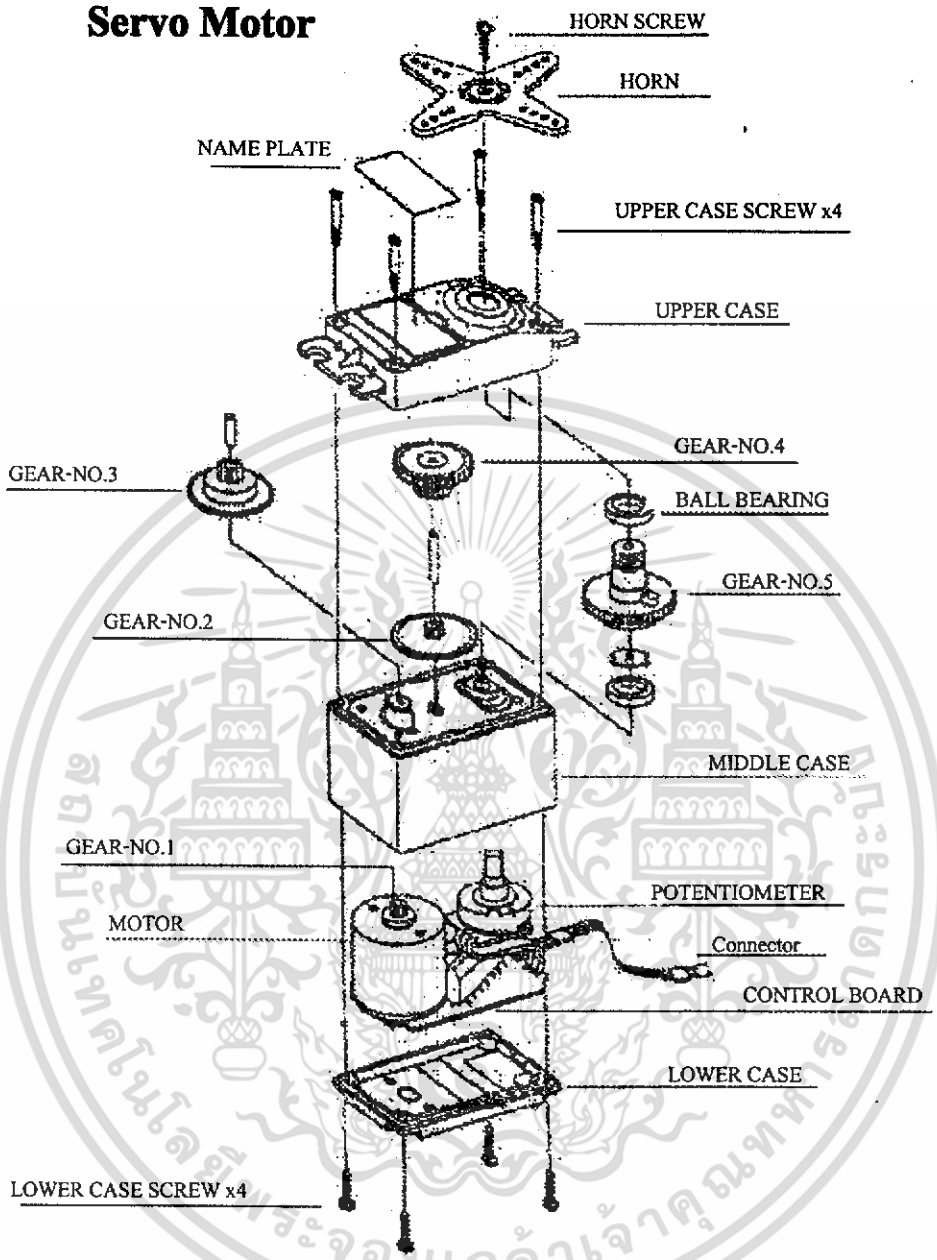
2.1.1 ความหมายของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์(Servo motor) คือ มอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ถูกประกอบรวมกับ ชุดเกียร์และส่วนควบคุมต่างๆ ไว้ ในโมดูลเดียวกันหรือภายในกล่องพลาสติกเดียวกัน ตามรูปที่ 2.1 โดยมอเตอร์ชนิดนี้จะมีสายต่อใช้งานเพียง 3 เส้นเท่านั้น คือ สายไฟระดับ 5 โวลต์ (VCC), สายไฟระดับ 0 โวลต์ (GND) และสายสัญญาณควบคุม (Control Line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้าย หรือขวาได้จากสายสัญญาณเพียงเส้นเดียว โดยสัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณพัลส์วิดมอด (PWM) แบบระดับแรงดัน TTL โดยระดับแรงดันที่จ่ายให้มอเตอร์นี้จะอยู่ในช่วงประมาณ 4 ถึง 6 โวลต์ ขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว ข้อดีของมอเตอร์ชนิดนี้ก็คือ จะมีขนาดเล็กน้ำหนักเบา, ให้แรงบิดสูง, กินพลังงานน้อย และสามารถควบคุมด้วยแรงดันลอจิกที่เป็น TTL ได้โดยตรงไม่จำเป็นต้องต่อวงจรขับอื่นๆ เพราะมอเตอร์ชนิดนี้จะมีวงจรควบคุมบรรจุไว้ภายในอยู่แล้ว ซึ่งมอเตอร์ชนิดนี้สามารถควบคุมให้หมุนไปในตำแหน่งหรือทิศทางองศาที่ต้องการได้ โดยอาศัยสัญญาณความกว้างพัลส์ที่ป้อนให้มอเตอร์ แต่เซอร์โวมอเตอร์นี้จะหมุนได้แค่เพียงในช่วงประมาณ 180 องศา หรือครึ่งรอบเท่านั้น หรือบางรุ่นอาจหมุนได้ถึง 210 องศา แต่จะไม่สามารถหมุนเป็นวงรอบได้ เนื่องจากโครงสร้างภายในจะประกอบด้วย ตัวต้านทานชนิดปรับค่าได้ (VR) ที่ทำหน้าที่ตรวจสอบตำแหน่งการหมุนของมอเตอร์ และตัวต้านทานนี้จะถูกยึดติดกับแกนหมุนของมอเตอร์ ซึ่งจากการที่ตัวต้านทานปรับค่านี้ไม่สามารถหมุนเป็นวงรอบได้ ดังนั้นเซอร์โวมอเตอร์จึงถูกออกแบบให้หมุนได้เพียงแค่ประมาณ 180 องศา หรือครึ่งรอบเท่านั้น เพื่อป้องกันความเสียหายที่จะเกิดกับตัวต้านทานปรับค่าได้ แต่ถ้าหากเราต้องการให้มอเตอร์หมุนเป็นวงรอบ (360 องศา) นั้นก็สามารถทำได้ โดยจะต้องทำการปรับแต่งดัดแปลงชิ้นส่วนบางอย่างของเซอร์โวมอเตอร์



รูปที่ 2.1 แผนภาพเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แผนภาพแสดงส่วนประกอบของเซอร์โวมอเตอร์

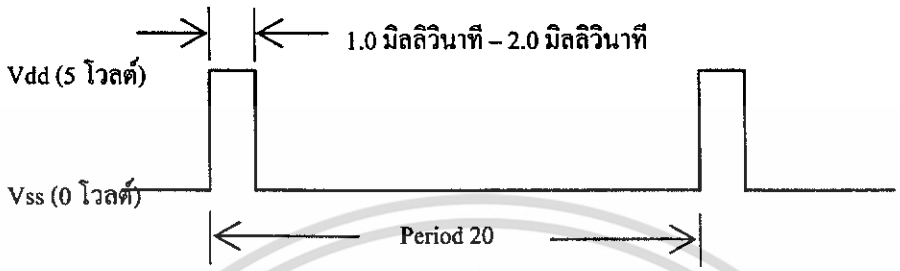
2.1.2 หลักการทำงานของเซอร์โวมอเตอร์

การควบคุมการทำงานของ เซอร์โวมอเตอร์ ทำได้โดยการป้อนสัญญาณความกว้างพัลส์ ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆ โดยทั่วไปดังแสดงไว้ในรูปที่ 2.3 รูปสัญญาณความกว้างพัลส์ และความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุด ดังแสดงไว้ในรูปที่ 2.4 ค่าความกว้างพัลส์และระยะองศาการหมุนของเซอร์โวมอเตอร์ คือ

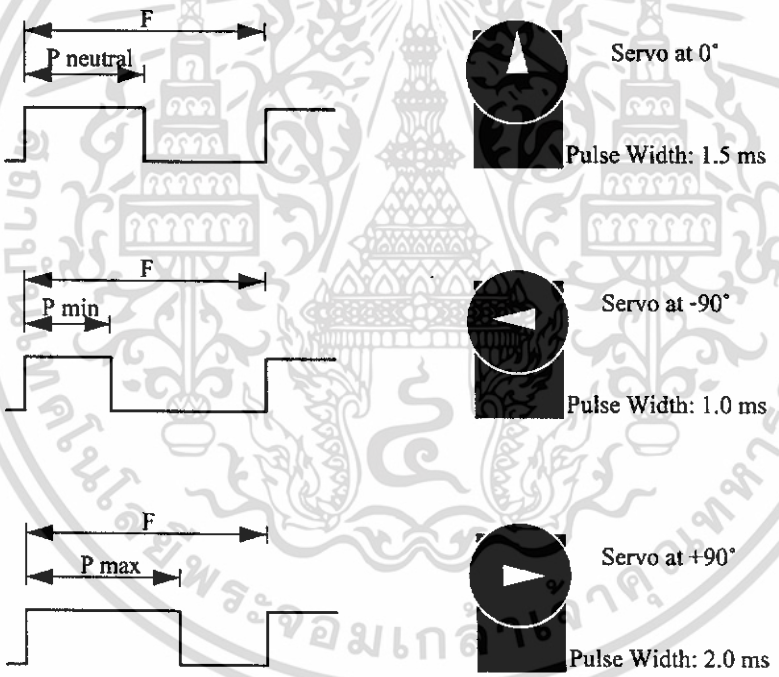
- สัญญาณความกว้างพัลส์ขนาด 1.5 มิลลิวินาที จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือ จุดกึ่งกลางของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สัญญาณความกว้างพัลส์ขนาด 1.0 มิลลิวินาที จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม -90 องศา หรือ ในทิศทางทวนเข็มนาฬิกา
- สัญญาณความกว้างพัลส์ขนาด 2.0 มิลลิวินาที จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม +90 องศา หรือ ในทิศทางตามเข็มนาฬิกา



รูปที่ 2.3 แผนภาพแสดงสัญญาณความกว้างพัลส์



รูปที่ 2.4 แผนภาพแสดงค่าความกว้างพัลส์และระยะองศาการหมุนของเซอร์โวมอเตอร์

โดยค่าความกว้างพัลส์ และระยะองศาการหมุนของมอเตอร์ที่อธิบายด้านบน นั้นเป็นเพียงค่าประมาณเท่านั้น ทั้งนี้ระยะการหมุน และ ขนาดของพัลส์ที่ควบคุมการทำงานของมอเตอร์ในแต่ละยี่ห้ออาจจะไม่เท่ากัน ดังนั้นในการใช้งานจึงควรศึกษารายละเอียดของมอเตอร์ในแต่ละรุ่นที่นำมาใช้ ซึ่งโดยปกติแล้วรายละเอียดต่างๆ ของมอเตอร์มักจะมีติดมากับตัวมอเตอร์นั้นๆ อยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเซอร์โวมอเตอร์ ยี่ห้อ GWS และ HITEC นั้นจะใช้ระบบเฟืองที่ต่างกันทำให้มีทิศทางการหมุนที่ต่างกัน โดยที่จะตรงข้ามกัน เช่น ส่งสัญญาณพัลส์ 1 มิลลิวินาที มอเตอร์ยี่ห้อ GWS จะหมุนทวนเข็มนาฬิกา ส่วนมอเตอร์ยี่ห้อ HITEC จะหมุนในทิศทางตามเข็มนาฬิกา เป็นต้น

ส่วนการที่จะควบคุมให้มอเตอร์หมุนเป็นมุมอื่นๆ นั้นก็สามารถทำได้โดยการป้อนสัญญาณพัลส์เป็นระดับความกว้างต่างๆ โดยอ้างอิงจากจุด ทั้ง 3 จุดที่กล่าวมานี้ ตัวอย่างเช่น ถ้าต้องการให้มอเตอร์หมุนไปที่มุม -45 องศา เราก็จะต้องป้อนสัญญาณพัลส์ที่มีความกว้าง 1.25 มิลลิวินาที เป็นต้น และ สัญญาณพัลส์นี้จะต้องจ่ายให้มอเตอร์ทุกๆ 20 มิลลิวินาที เพื่อรักษาสภาพตำแหน่งของมอเตอร์ไว้

โดยหลักการก็คือ จะอาศัยการเปรียบเทียบช่วงเวลาของความกว้างพัลส์ที่จ่ายให้กับมอเตอร์ทางขาสัญญาณควบคุมกับค่าเวลาวงจรตัวต้านทานและตัวเก็บประจุ (Resistance Capacitance; RC) ภายในบอร์ดควบคุมในตัวของมอเตอร์ ซึ่งค่าเวลาของวงจร RC นี้จะมีการเปลี่ยนแปลงตามการหมุนของมอเตอร์ เนื่องจากตัวต้านทานปรับค่าจะถูกยึดติดอยู่กับแกนหมุนของมอเตอร์ ซึ่งการหมุนของมอเตอร์จะทำให้ค่าความต้านทานของตัวต้านทานปรับค่าได้ (Variable Resistance; VR) เปลี่ยนแปลงไป เป็นผลทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงตามไปด้วย โดยในขณะที่เราป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ทางขาสัญญาณควบคุม สัญญาณนี้จะถูกนำไปเปรียบเทียบกับค่าเวลาของวงจร RC หากค่าทั้ง 2 ไม่เท่ากันมอเตอร์ก็จะหมุนทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงจนกระทั่งค่าเวลาความกว้างพัลส์ของ วงจร RC เปลี่ยนแปลงจนเท่ากับสัญญาณพัลส์ทางขาควบคุม (Control line) มอเตอร์จึงจะหยุดหมุน

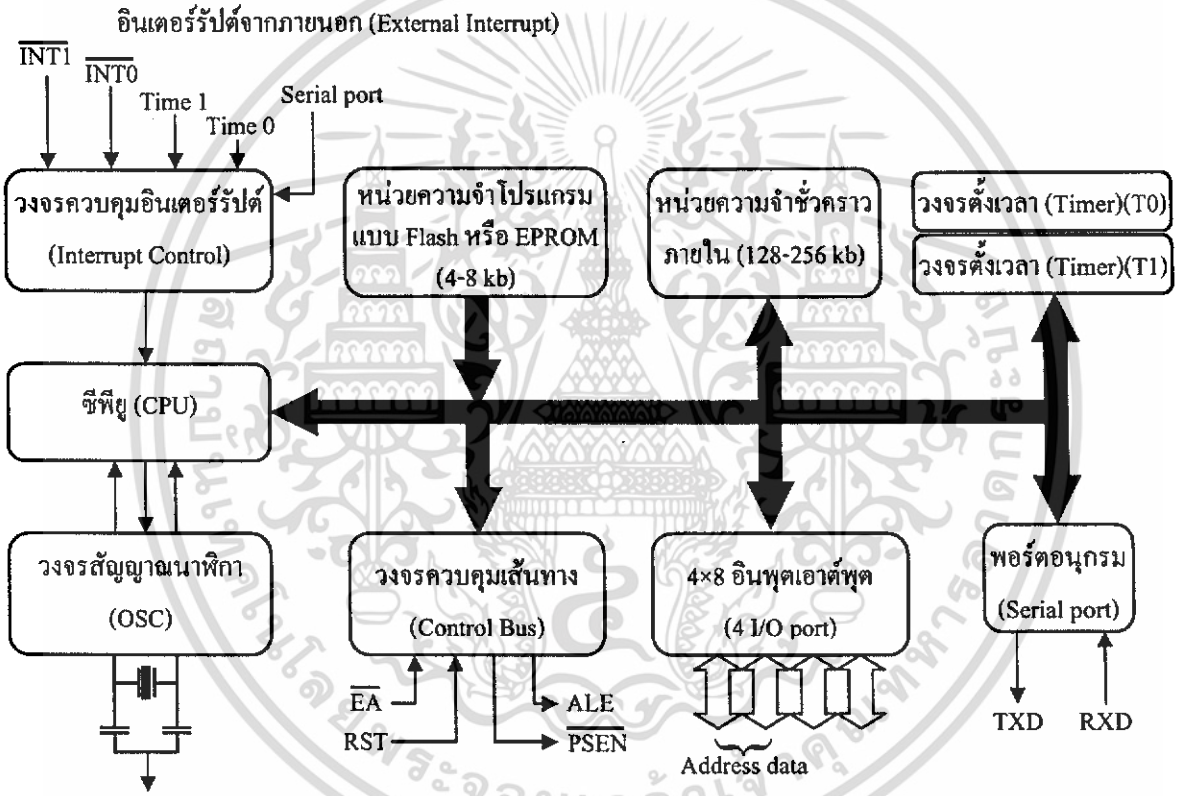
2.2 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 (AT89C51)

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นไมโครคอนโทรเลอร์ของบริษัทอินเทล (Intel Cooperation) ถูกผลิตขึ้นและได้มีการนำไปใช้งานอย่างแพร่หลายตั้งแต่ปี ค.ศ. 1980 เป็นต้นมา และได้มีบริษัทต่างๆ หลายบริษัทได้รับลิขสิทธิ์ ในการผลิต, จำหน่าย และได้รับการพัฒนาเพื่อเพิ่มประสิทธิภาพการทำงานในด้านต่างๆ มากขึ้นและหนึ่งในจำนวนนั้นก็มีบริษัทเอทเมล (ATMEL) ได้พัฒนาชิปตระกูล MCS-51 ขึ้นโดยใช้เทคโนโลยีของแฟลส เมมโมรี (Flash memory) ซึ่งต่างจากของ Intel ที่ใช้หน่วยความจำถาวรแบบลบได้ (Erasable – Programmable Read only Memory; EPROM) และหน่วยความจำแบบถาวร (Read only Memory; ROM) ซึ่งข้อดีของแฟลส เมมโมรี (Flash memory) ก็คือทำให้สามารถเขียนโปรแกรมและสามารถลบทับลงไปอย่างง่าย

ในปัจจุบันเทคโนโลยีระบบฝังตัว (embedded system) ได้เข้ามามีบทบาทอย่างมาก อุปกรณ์ไฟฟ้าอิเล็กทรอนิกส์ต่าง ๆ ได้มีการเพิ่มหน่วยประมวลผลเข้าไปภายในตัวมัน ในขณะที่เทคโนโลยีหน่วยประมวลผลได้มีการพัฒนาอยู่ตลอดเวลา โดยทั่วไปแล้ววงจรควบคุมการทำงานอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ อาจอยู่ในรูปของวงจรดิจิตอลทั่ว ๆ ไปก็ได้ แต่ในปัจจุบันจะพบว่าระบบฝังตัวจะใช้ไมโครคอนโทรลเลอร์มากขึ้น เนื่องจากมีความยืดหยุ่นสูงทำให้การควบคุมการทำงานจะเน้นไปที่การทำงานทางซอฟต์แวร์ (software) แทนการพัฒนาทางด้านฮาร์ดแวร์ (hardware) เช่นในอดีต ส่วนใหญ่ผู้พัฒนาเทคโนโลยีระบบฝังตัวจะให้ความสำคัญกับไมโครโพรเซสเซอร์หรือไมโครคอนโทรลเลอร์เป็นหลัก การแบ่งระดับความซับซ้อนของระบบฝังตัวสามารถแบ่งได้หลายระดับ ระบบฝังตัวที่มีความซับซ้อนมาก ๆ จะใช้ไมโครโพรเซสเซอร์ที่มีความสามารถสูงเป็นหน่วยประมวลผลกลาง เช่น ตระกูล x86 ของบริษัทอินเทล สำหรับระบบฝังตัวขนาดเล็กจะใช้ไมโครคอนโทรลเลอร์ที่มีการประมวลผลแบบ 4 บิต หรือ 8 บิต ไมโครคอนโทรลเลอร์ที่เหมาะสมกับการศึกษาเบื้องต้นตัวหนึ่งก็คือ ไมโครคอนโทรลเลอร์ตระกูล MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปี ค.ศ. 1981 บริษัทอินเทลได้เปิดตัวไมโครคอนโทรลเลอร์แบบ 8 บิต โดยให้ชื่อว่า 8051 โดยไมโครคอนโทรลเลอร์ตัวนี้จะมีหน่วยความจำชั่วคราว (Random Access Memory; RAM) ขนาด 128 ไบต์ หน่วยความจำประเภท ROM ขนาด 4 กิโลไบต์ มีพอร์ตขนานขนาด 8 บิต จำนวน 4 พอร์ต มีไทเมอร์ 2 ตัว และพอร์ตอนุกรม 1 พอร์ต โดยทั้งหมดจะรวมอยู่ภายในชิปเพียงชิปเดียว ไดอะแกรมภายในตัวมันแสดงได้ ดังรูป 2.4 และเนื่องจากหน่วยความจำโปรแกรมเป็นแบบ ROM ที่อยู่ภายในตัวมันดังนั้นการโปรแกรมการทำงานต้องโปรแกรมมาจากโรงงานที่ผลิตชิปนี้ออกมาโดยตรงนอกจากนี้บริษัทอินเทลยังได้ผลิตไมโครคอนโทรลเลอร์ออกมาอีกหลายเบอร์ที่มีโครงสร้างภายในใกล้เคียงกับ 8051 โดยเรียกรวมๆ ว่า ตระกูล MCS-51 หรือตระกูล 51 โดยการมีการปรับปรุงโครงสร้างภายในและเพิ่มองค์ประกอบต่างๆ เข้าไปอีก และยังมีบริษัทอื่นๆ ผลิตตามออกมาด้วย แต่จะใช้เบอร์ที่เรียกแตกต่างกันออกไป



รูปที่ 2.5 แผนภาพแสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์

2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51

1) เป็นชิปขนาด 40 ขา
2) มีหน่วยความจำโปรแกรมชนิดแฟลชเมโมรี หรือชนิดที่เขียนและลบได้รวดเร็ว ขนาด 4 กิโลไบต์ทนต่อการเขียนลบได้ 1000 ครั้ง และคงค่าข้อมูลไว้ได้นาน 10 ปี
3) ใช้แรงไฟตรง ตั้งแต่ 2.7 โวลต์ - 6 โวลต์
4) ความถี่ ออสซิสเลเตอร์ (Oscillator) 0 -- 24 เมกะเฮิรตซ์
5) หน่วยความจำโปรแกรมสามารถล็อคโปรแกรม (Program Memory Lock) ได้ 3 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) มีหน่วยความจำแรม (Ram) ภายในตัวชิปอยู่ 128 ไบต์

7) มีขาอินพุต-เอาท์พุต (Input/Output Pin) จำนวน 32 ขา

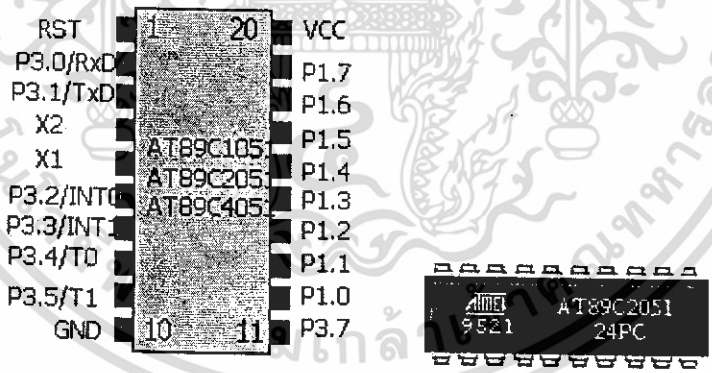
8) ตัวติดตั้ง/ตัวนับเวลาขนาด 16 ไบต์ 2 ตัว

9) สามารถรับการอินเตอร์รับ (Interrupt) หรือสัญญาณขัดจังหวะให้มีการทำงานก่อนได้จาก 5 แหล่ง

10) ช่องโปรแกรมอนุกรม UART (Programmable Serial UART Channel)

ความเร็วในการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51 การให้ไมโครคอนโทรลเลอร์ทำงานนั้น จะต้องโปรแกรมให้กับตัวมันก่อน การวัดความเร็ว ในการทำคำสั่งของโปรแกรมจะดูจากรอบสัญญาณนาฬิกา หรือที่เรียกว่า แมชชีนไซเคิล ซึ่งในตารางคำสั่งของไมโครคอนโทรลเลอร์แต่ละตัว จะมีข้อมูลบอกไว้ว่าการทำคำสั่งแต่ละคำสั่ง จะใช้สัญญาณนาฬิกากี่แมชชีนไซเคิล สำหรับไมโครคอนโทรลเลอร์ MCS-51 เบอร์ที่เป็นมาตรฐานนั้น 1 แมชชีนไซเคิลจะใช้สัญญาณนาฬิกา 12 ลูก ดังนั้นถ้า MCS-51 ทำงาน ที่สัญญาณนาฬิกา 12 เมกะเฮิรตซ์ แล้ว การทำงาน 1 ไซเคิล จะมีค่าเท่ากับ 1 ไมโครวินาที หรือมีความเร็วในการทำงาน 1 เมกะเฮิรตซ์ ถ้าหากต้องการให้ MCS-51 ตัวนั้นทำงานได้เร็วขึ้นจะต้องเพิ่มสัญญาณนาฬิกาให้กับมัน

โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์8051 ประกอบด้วยอุปกรณ์ต่าง ๆ ดังนี้ ส่วนของหน่วยความจำภายในสำหรับเก็บข้อมูลขนาด 128 ไบต์ (Internal Data Memory) ส่วนของหน่วยความจำภายในสำหรับเก็บโปรแกรมที่มีขนาด 4 กิโลไบต์ (Internal Program Memory) อุปกรณ์ควบคุมการอินเตอร์รัพท์ (Interrupt Control Unit) ตัวตั้งเวลาและตัวนับเวลาขนาด 16 บิต 2 ชุด (Timer/Counter 0 and Timer/Counter 1) พอร์ตควบคุมการสื่อสารอนุกรมแบบ Full Duplex ซึ่งสามารถรับส่งข้อมูลพร้อมกันได้ พอร์ตขนานสำหรับติดต่อกับอุปกรณ์ภายนอกจำนวน 4 พอร์ต พอร์ตละ 8 บิต วงจรผลิตสัญญาณนาฬิกาภายใน



รูปที่ 2.6 แผนภาพแสดงสัญลักษณ์ของไมโครคอนโทรลเลอร์เบอร์ AT89C2051

2.2.2 หน่วยความจำ (Memory)

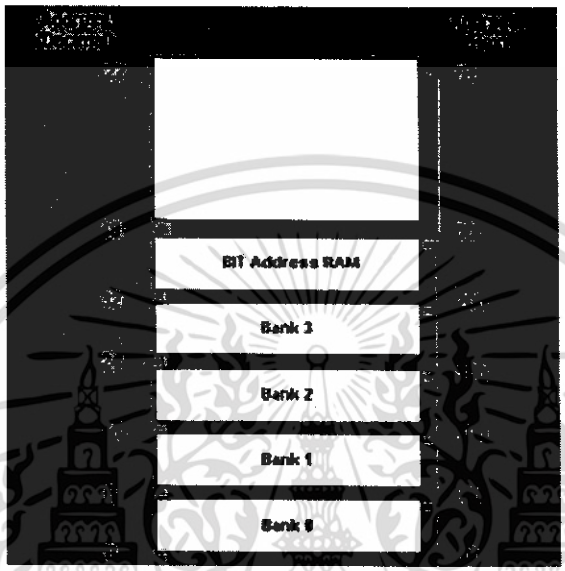
ไอซีในตระกูล MCS-51 แบ่งหน่วยความจำออกเป็น 2 ส่วน คือ หน่วยความจำโปรแกรม (Program Memory) และหน่วยความจำข้อมูล (Data Memory)

1) หน่วยความจำโปรแกรม (Program memory) ของไอซีตระกูลMCS-51 สามารถอ้างได้ถึง 64 กิโลไบต์ สำหรับไอซี AT89C2051 มีหน่วยความจำขนาด 4 กิโลไบต์ แบบภายใน (Internal Program Memory) ไม่สามารถต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มเติมจากภายนอกได้อีก และหน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ เป็นแบบแฟลส เมมโมรี สามารถโปรแกรมได้ประมาณ 1000 ครั้ง (การโปรแกรมต้องอาศัยอุปกรณ์ที่ออกแบบมาสำหรับโปรแกรม)

2) หน่วยความจำข้อมูล (Data memory) ของไอซี AT89C2051 มีขนาด 128 ไบต์ (00H – 7FH) และอีก 128 ไบต์ถัดไป (80H – FFH) เป็นส่วนของรีจิสเตอร์หน้าที่พิเศษ (Special Function Register หรือ SFR) ไม่สามารถใช้คำสั่ง MOV X (External Data Memory Access) เพื่ออ่านหรือเขียนหน่วยความจำจากภายนอกได้



รูปที่ 2.7 แผนภาพแสดงพื้นที่หน่วยความจำภายในของไอซี AT89C2051

จากรูปที่ 2.7 จะเห็นได้ว่ามีหน่วยความจำข้อมูลภายในตั้งแต่ 00 H - 7F H ถูกแบ่งออกเป็น 3 ส่วนตามประเภทการใช้งาน ซึ่งจะมีส่วนเกี่ยวข้องกับการทำงานของไมโครคอนโทรลเลอร์ ในส่วนของการกำหนดการทำงานของรีจิสเตอร์ รีจิสเตอร์ที่สามารถเข้าถึงได้ในระดับบิต และพื้นที่ที่ใช้เป็นหน่วยเก็บความจำภายใน ดังนี้

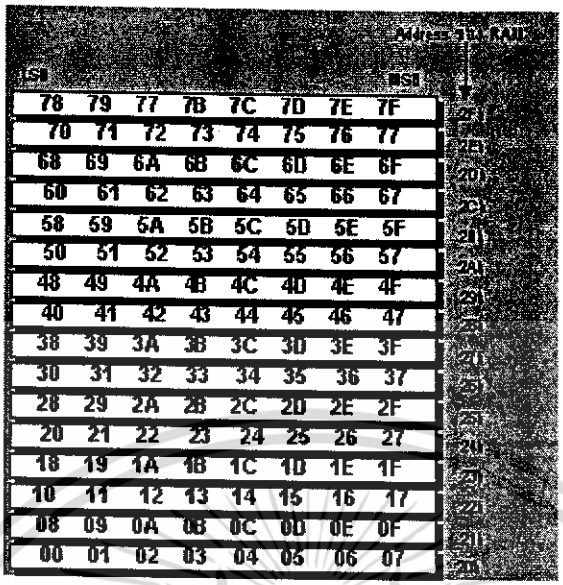
1) 00H – 1FH จำนวน 32 ไบต์ ถูกแบ่งเป็นกลุ่มหรือแบงก์ (Bank) ขนาด 8 ไบต์ จำนวน 4 กลุ่ม ถูกใช้งานเป็นรีจิสเตอร์เรียกว่า R0-R7 และสามารถเลือกแบงก์ ได้จากรีจิสเตอร์ PSW (Program Status Word) ดังตารางที่ 2.1

ตารางที่ 2.1 รีจิสเตอร์ RO - R7

แอดเดรส	รีจิสเตอร์แบงก์	ชื่อรีจิสเตอร์ใช้งาน
00H-07H	0	R0-R7
08H-0FH	1	R0-R7
10H-17H	2	R0-R7
18H-1FH	3	R0-R7

2) 20H – 2FH ขนาด 16 ไบต์ ส่วนนี้สามารถอ้างถึงข้อมูลได้ทั้งแบบไบต์ และแบบบิตดังนั้นถ้ามองการอ้างอิงในลักษณะบิตแล้วก็จะอ้างได้ถึง 128 บิต ดังรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แผนภาพแสดงหน่วยความจำภายในบริเวณที่อ้างถึงแบบบิตได้

3) 30H – 7FH เป็นส่วนของหน่วยความจำข้อมูลที่อ้างในลักษณะไบต์เท่านั้น สามารถนำไปใช้งานได้อย่างอิสระ

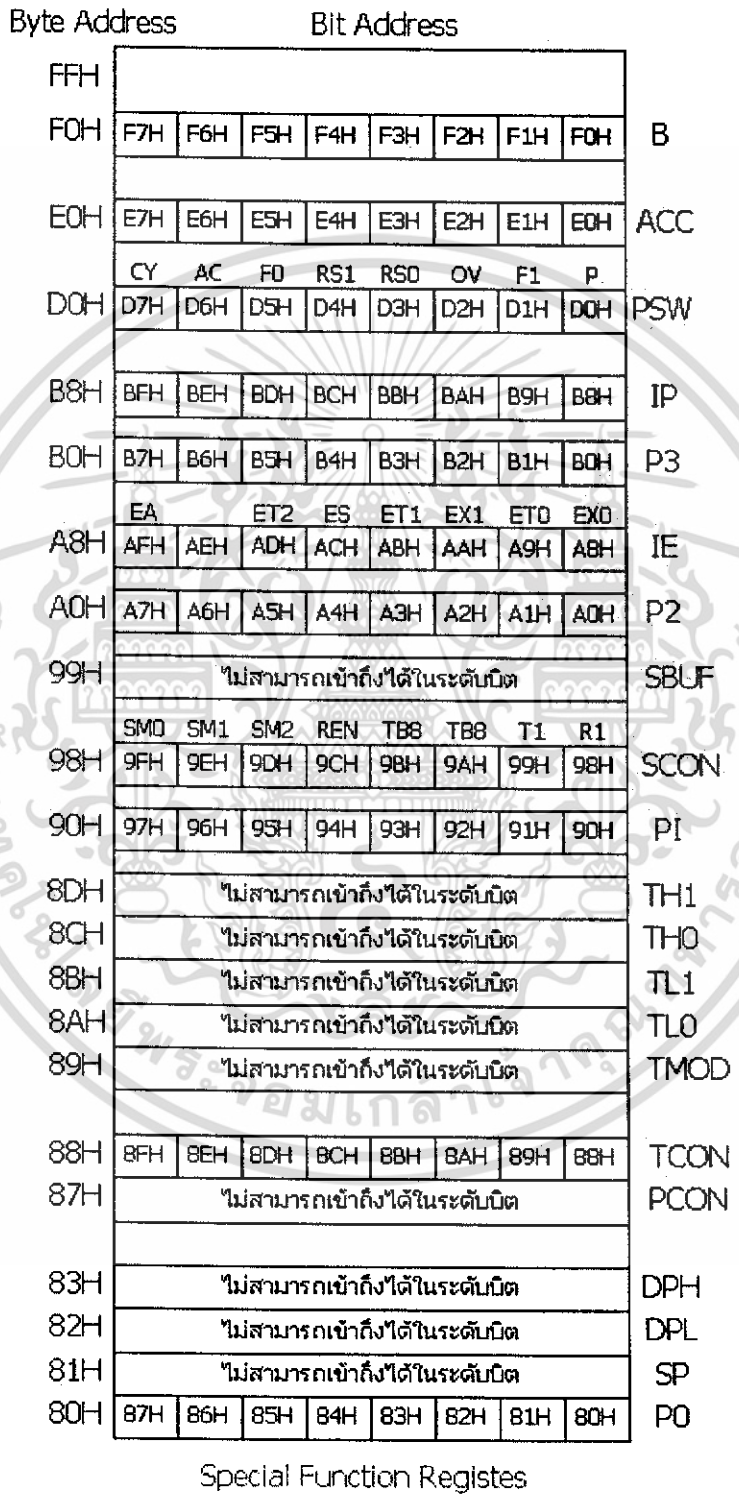
4) 80H – FFH เป็นส่วนของหน่วยความจำที่นำมาใช้เป็นรีจิสเตอร์พิเศษ (SFR) ไม่สามารถนำไปใช้งานปกติได้ เนื่องจาก SFR เป็นรีจิสเตอร์ที่ใช้สำหรับควบคุมหน้าที่การทำงานของอุปกรณ์และพอร์ตต่างๆ ตามตารางที่ 2.2

ตารางที่ 2.2 รีจิสเตอร์หน้าที่พิเศษ (SFR)

ชื่อรีจิสเตอร์	คำจำกัดความ	ความสามารถอ้างถึงบิต
ACC	Accumulator	ได้
B	B register	ได้
PSW	Program status Word	ได้
SP	Stack Pointer	ได้
DPTR	Data Pointer(DPH &DPL)	ได้
P1	Port1	ได้
P3	Port3	ได้
IE	Interrupt Enable	ได้
TMOD	Timer/Counter mode	ไม่ได้
TCON	Timer/Counter control	ได้
SCON	Serial Control	ไม่ได้
SBUF	Serial Data Buffer	ไม่ได้
PCON	Power Control	ไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์เฉพาะหรือรีจิสเตอร์พิเศษ (Special Function Register) ที่ใช้ในการควบคุมการทำงานของไมโครคอนโทรลเลอร์ จะอยู่ในหน่วยความจำตำแหน่งแอดเดรสที่ 80H – FFH ซึ่งสามารถจะเรียกใช้ชื่อของรีจิสเตอร์ได้โดยตรง หรืออาจจะเรียกชื่อตามตำแหน่งแอดเดรสก็ได้ รีจิสเตอร์เฉพาะจะประกอบด้วยดังแสดงในรูปที่ 2.9



รูปที่ 2.9 แผนภาพแสดงการจัดหน่วยความจำและตำแหน่งของรีจิสเตอร์เฉพาะ (Special Function Register)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 พอร์ตอินพุตและพอร์ตเอาท์พุต

1) พอร์ต 1 ขนาด 8 bit (P1.0– 1.7) แบบ Bidirectional สามารถขับแอลอีดีได้โดยตรง โดย P1.2- P1.7 จะมี Pull – up อยู่ภายใน ส่วน P1.0 และ P1.1 ใช้เป็น Analog comparator

2) พอร์ต 2 (P3.0 – P3.5) เป็น Bi – direction และมี R pull –up ภายในส่วน P3.6 จะไม่มี Pin แต่จะเป็นเอาท์พุต ของ Analog comparator (ดูจาก Block diagram) นอกจากนั้น P3.0- P3.5 ยังทำหน้าที่พิเศษดังต่อไปนี้

P3.0 RXD (Serial input port)	P3.1 TXD (Serial input port)
P3.2 INT0 (External interrupt 0)	P3.3 INT1 (External interrupt 1)
P3.4 T0 (Timer 0 external input)	P3.5 T1 (Timer 1 external input)

2.2.4 การทำงานของรีจิสเตอร์

2.2.4.1 รีจีสเตอร์ A (Accumulator = Acc)

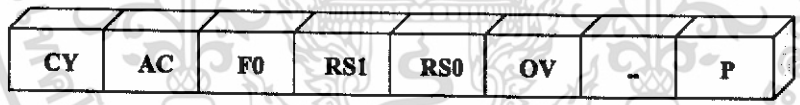
เป็นรีจีสเตอร์ขนาด 8 บิต สามารถเข้าถึงได้ระดับบิต ทำหน้าที่หลักในการคำนวณทางคณิตศาสตร์และลอจิก เก็บผลลัพธ์จากการประมวล และสามารถเก็บข้อมูลขนาด 8 บิตได้ อยู่ที่ตำแหน่งหน่วยความจำ E0H

2.2.4.2 รีจีสเตอร์ B (B Register)

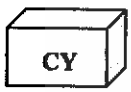
เป็นรีจีสเตอร์ขนาด 8 บิต ใช้เก็บข้อมูลทั่วไป และยังทำหน้าที่พิเศษคือ เก็บข้อมูลตัวกระทำการคูณหรือการหาร เช่น รีจีสเตอร์ A เก็บค่าตัวตั้ง และรีจีสเตอร์ B เป็นตัวคูณหรือหาร สามารถเข้าถึงได้ระดับ อยู่ที่ตำแหน่งหน่วยความจำ 0F0H

2.2.4.3 รีจีสเตอร์ PSW (Program Status Word)

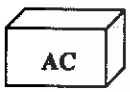
ทำหน้าที่เก็บสภาวะการงานของคำสั่ง มีขนาด 7 บิต อยู่ที่ตำแหน่งหน่วยความจำ 0D0H ซึ่งมีการเก็บสภาวะการทำงานของแต่ละบิตดังต่อไปนี้



รูปที่ 2.10 แผนภาพแสดงรีจีสเตอร์ PSW (Program Status Word)

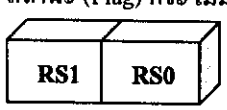


(Carry Flag) ทำหน้าที่เก็บค่าตัวทดเมื่อมีการกระทำทางคณิตศาสตร์และลอจิก ถ้าผลลัพธ์จากการประมวลผลที่ได้มีค่าเกิน 8 บิต (0FFH) บิตที่เกินก็จะเก็บไว้ที่บิต CY ซึ่งเป็นบิตทด



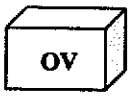
(Auxiliary Carry Flag) เป็นตัวช่วยเมื่อมีการคำนวณ และเกิดการยืมค่าหรือเกิดทดขึ้นระหว่างบิตที่ 3 กับบิตที่ 4 ของค่ารีจีสเตอร์ บิตนี้จะมีค่าลอจิกเป็น "1" ส่วนมากใช้กับระบบเลขฐานสอง (Binary Code Decimal)

(Flag 0) ทำหน้าที่เป็นตัวเก็บสถานะทั่วไป ถ้ามีการตั้งค่าสถานะนี้ไว้ เมื่อมีการกระทำตามคำสั่งที่มีผลต่อสถานะ (Flag) ก็จะไม่มีผลกระทบต่อค่าที่กำหนดไว้

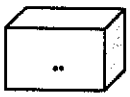


(Register Select 1, 0) ทำหน้าที่เลือกใช้ตำแหน่งของรีจีสเตอร์ R0-R7 มีอยู่ 4 ตำแหน่ง ตำแหน่งละ 8 ตัว (R0-R7)

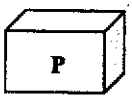
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(Overflow) เป็นค่าแสดงสถานะเมื่อมีการคำนวณหรือการกระทำทางลอจิก ผลลัพธ์ที่ได้จะมีค่าเกินกว่าที่ตำแหน่งหน่วยความจำจะรับได้ ทำให้สถานะนี้เป็น "1" และยังสามารถใช้เป็นค่าแสดงผลลัพธ์ที่เป็นลบ



บิตนี้ได้ใช้งาน ผู้ใช้สามารถนำไปใช้งานได้



(Parity Flag) ใช้ตรวจสอบค่าที่เก็บอยู่ในรีจีสเตอร์ A แต่ละบิตมีค่าบิตเป็น 1 รวมกันเป็นจำนวนคู่หรือคี่ ถ้ามีจำนวนเป็นคู่ ค่าสถานะนี้จะเป็น "1" แต่ถ้าเป็นจำนวนคี่จะมีค่าสถานะเป็น "0"

2.2.4.4 รีจีสเตอร์สแตก (SP = Stack Pointer)

เป็นรีจีสเตอร์ขนาด 8 บิต มีตำแหน่งอยู่ที่ 81H ใช้เก็บตำแหน่งตัวชี้ค่าสแตก (เข้าที่หลังออกก่อน) และใช้เก็บค่าตำแหน่งของโปรแกรมหลัก เมื่อไมโครคอนโทรเลอร์กระโดดไปทำงานโปรแกรมย่อยเสร็จ ไมโครคอนโทรลเลอร์จะกลับมาทำงานที่โปรแกรมหลักอีกที่ ก่อนที่จะกระโดดไป ไมโครคอนโทรลเลอร์จะเก็บค่าตำแหน่งไว้ที่สแตกก่อน ในกรณีการเขียนโปรแกรมย่อยหลายๆ โปรแกรมค่าสแตกก็จะมีการเก็บค่าตำแหน่งมากขึ้น พื้นที่ที่ต้องใช้งานก็เพิ่มมากขึ้น ค่ารีจีสเตอร์สแตกสามารถกำหนดตำแหน่งใหม่ได้แต่ถ้าไม่กำหนดตำแหน่งไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะกำหนดตำแหน่งไว้ที่ 07H

2.2.4.5 รีจีสเตอร์เป็นตัวนับขั้นตอนในโปรแกรม (PC = Program Counter)

เป็นรีจีสเตอร์ขนาด 16 บิต มีหน้าที่บอกตำแหน่งการทำงานของไมโครคอนโทรลเลอร์แต่ละคำสั่งจะมีขนาดความยาวโค้ดคำสั่งต่างกัน ซึ่งมีความสำคัญมากในการตรวจสอบการเขียนของโปรแกรมให้ไมโครคอนโทรลเลอร์ประมวลผลและปฏิบัติตามลำดับขั้นตอนแผนงานที่ได้วางไว้ รีจีสเตอร์ตัวนับขั้นตอนในโปรแกรมจะเป็นตัวเก็บค่าตำแหน่งทุกตำแหน่งที่ไมโครคอนโทรลเลอร์ทำงานอยู่ปัจจุบัน

2.2.4.6 รีจีสเตอร์ตัวชี้ข้อมูล (DPTR = Pointer)

เป็นรีจีสเตอร์ขนาด 8 บิต 2 ชุด ประกอบด้วยรีจีสเตอร์ไบต์ต่ำ (DPH) อยู่ที่ตำแหน่ง 82H และ 83H สามารถใช้ได้ 8 บิต และใช้รวมเป็นขนาด 16 บิต ทำหน้าที่กำหนดตำแหน่งข้อมูลในหน่วยความจำและอุปกรณ์ภายนอกที่ไมโครคอนโทรลเลอร์ต้องการติดต่อ

2.2.4.7 รีจีสเตอร์พอร์ต (Port Register)

ไมโครคอนโทรลเลอร์ MCS-51 ได้กำหนดให้ติดต่อทุกอย่างกระทำผ่านพอร์ต ซึ่งมีอยู่ 4 พอร์ต มีขนาดพอร์ต 8 บิต คือ พอร์ต 0 (P0 = 80H) พอร์ต 1 (P1 = 90H) พอร์ต 2 (P2 = A0H) และพอร์ต 3 (P3 = B0H) ทุกพอร์ตสามารถใช้อินพุตและเอาต์พุตที่ควบคุมได้ระดับบิต

2.2.4.8 รีจีสเตอร์บัฟเฟอร์ข้อมูลอนุกรม (SBUF = Serial Data Buffer)

เป็นรีจีสเตอร์ที่ทำหน้าที่รับส่งข้อมูลแบบอนุกรม มีขนาด 8 บิต อยู่ที่ตำแหน่ง 99H ซึ่งปกติใช้การติดต่อข้อมูลอนุกรมมีบัฟเฟอร์ 2 ชุด คือ บัฟเฟอร์สำหรับส่งข้อมูล (transmit buffer register) ผ่านไปยังขา TXD และบัฟเฟอร์สำหรับรับข้อมูล (receive buffer register) ผ่านมายังขา RXD เมื่อต้องการส่งข้อมูลอนุกรมก็สามารถเขียนข้อมูลไปยังรีจีสเตอร์ SBUF และข้อมูลจะถูกส่งไปยังบัฟเฟอร์เพื่อให้ไมโครคอนโทรลเลอร์ส่งข้อมูลออกไปภายนอก

2.2.4.9 รีจีสเตอร์เกี่ยวกับเวลา (Timer Register)

เป็นรีจีสเตอร์ขนาด 16 บิต แบ่งเป็นรีจีสเตอร์คู่ คือ ไบต์ต่ำและไบต์สูง ใช้เก็บค่าของตัวนับเวลา (Counter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในไมโครคอนโทรลเลอร์ เพื่อใช้เป็นฐานเวลาหรือจับเวลานับจำนวนสัญญาณนาฬิกา (Pulse) ในไมโครคอนโทรลเลอร์ตระกูล MCS-51 ประกอบด้วย TH0 TH1, TL0 TL1 หรือ TL2 TH2

2.2.4.10 รีจีสเตอร์แคปเจอร์ (Capture Register)

เป็นรีจีสเตอร์ขนาด 16 บิต คือ RCAP2L ไบต์ต่ำ และ RCAP2H ไบต์สูง ใช้ร่วมกับ TL2 TH2 ทำหน้าที่ให้ไมโครคอนโทรเลอร์ตรวจจับเวลาการเปลี่ยนแปลงสถานะลอจิกที่ขา T2EX เพื่อใช้วัดคาบเวลา ความถี่ และการเปลี่ยนแปลงสัญญาณนาฬิกาที่ขา T2EX

2.2.4.11 รีจีสเตอร์ควบคุม (Control Register)

เป็นรีจีสเตอร์ที่ใช้ควบคุมการทำงานส่วนต่างๆ ของไมโครคอนโทรเลอร์ ซึ่งประกอบด้วยรีจีสเตอร์ต่อไปนี้

2.2.4.12 รีจีสเตอร์ PCON (Power Control Register)

ทำหน้าควบคุมการทำงานของไมโครคอนโทรลเลอร์ โดยการกำหนดให้สัญญาณนาฬิกาหยุดทำงาน ทำให้ส่วนต่างๆ ภายในไมโครคอนโทรลเลอร์หยุดทำงานด้วย ซึ่งเป็นการลดพลังงานเมื่อไม่ต้องการให้ไมโครคอนโทรลเลอร์ทำงาน (Sleep mode)

2.2.4.13 รีจีสเตอร์ SCON (Serial Control Register)

เป็นรีจีสเตอร์ที่ทำหน้าที่ควบคุมวงจรการสื่อสารแบบอนุกรมภายในไมโครคอนโทรลเลอร์

2.2.4.14 รีจีสเตอร์ TCON, T2CON และ TMOD, T2MOD

ใช้ควบคุมการทำงานของวงจรจับเวลา/นับเวลาภายในไมโครคอนโทรลเลอร์

2.2.4.15 รีจีสเตอร์ IE และ IP (Interrupt Enable Control และ Interrupt Priority Control)

เป็นรีจีสเตอร์ที่ควบคุมการอินเตอร์รัปต์ของไมโครคอนโทรลเลอร์ มี IE (Interrupt Enable) เป็นรีจีสเตอร์สนองการทำงานของสัญญาณอินเตอร์รัปต์ และรีจีสเตอร์ IP (Priority Interrupt) ทำหน้าที่จัดลับดับความสำคัญของสัญญาณตอบสนองการอินเตอร์รัปต์ของไมโครคอนโทรลเลอร์ของไมโครคอนโทรลเลอร์

ในกรณีเริ่มต้นการทำงานใหม่ของไมโครคอนโทรลเลอร์หรือรีเซตเครื่อง ค่ารีจิสเตอร์ต่างๆ ที่อยู่ภายในไมโครคอนโทรลเลอร์จะถูกเซตค่าใหม่ ดังตารางที่ 2.3

ตารางที่ 2.3 ค่ารีจีสเตอร์ต่างๆ เมื่อเริ่มต้นทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51

สัญลักษณ์	ชื่อ	เลขฐานสอง
*Acc	Accumulator	00000000
*B	B Register	00000000
*PSW	Program Status Word	00000000
SP	Stack Pointer	00000111
DPTR	Data Pointer 2 ไบต์ (DPH , DPL)	
DPL	ไบต์ต่ำ (Low byte)	00000000
DPH	ไบต์สูง (High byte)	00000000
*P0	Port 0	11111111
*P1	Port 1	11111111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ค่ารีจีสเตอร์ต่างๆ เมื่อเริ่มต้นทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 (ต่อ)

สัญลักษณ์	ชื่อ	เลขฐานสอง
*P3	Port 2	11111111
*IP	Port 3	11111111
*IE	Interrupt Priority Control	8051 xxx00000 , 8052 xx00000
TMOD	Interrupt Enable Control	8051 0xx00000 , 8052 xx00000
*TCON	Timer /Counter Mode Control	00000000
*+T2CON	Timer /Counter Control	00000000
TH0	Timer /Counter 2 Control	00000000
TH1	Timer /Counter 1 High byte	00000000
TL	Timer /Counter 1 Low byte	00000000
+TH2	Timer /Counter 2 High byte	00000000
+TL2	Timer /Counter 2 Low byte	00000000
+RCAP2H	T/C 2 Capture Reg. High byte	00000000
+RCAP2L	T/C 2 Capture Reg. Low byte	00000000
*SCON	Serial Control	00000000
SBUF	Serial Data Buffer	ไม่ได้กำหนด
PCON	Power Control	0xxxxxxx

หมายเหตุ * รีจีสเตอร์ที่สามารถเข้าถึงระดับบิต + มีใช้เฉพาะรุ่น 80x52 (AT89xx52) x ไม่ได้กำหนด

2.2.5 ชุดคำสั่งของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีชุดคำสั่งทั้งหมด 111 คำสั่ง และแบ่งหน้าที่การทำงานออกเป็นกลุ่มตามลักษณะของงานที่ทำเพื่อให้ผู้เรียนสามารถศึกษาได้สะดวกขึ้น ซึ่งประกอบด้วยกลุ่มคำสั่งต่อไปนี้

1. กลุ่มคำสั่งทางคณิตศาสตร์ (Arithmetic instruction)
2. กลุ่มคำสั่งการกระทำลอจิก (Logical instruction)
3. กลุ่มสั่งการเคลื่อนย้ายข้อมูล (data transfer instruction)
4. กลุ่มคำสั่งการจัดการข้อมูลระดับบิต (bit manipulated instruction)
5. กลุ่มคำสั่งกระโดด (branch instruction)

- **รูปแบบการเขียนคำสั่ง**

ภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ MCS-51 จะมีรูปแบบมาตรฐานคล้ายกับ ไมโครคอนโทรลเลอร์ทั่วๆ ไป ประกอบด้วย 2 ส่วนใหญ่ๆ คือ ส่วนที่เป็นรหัสคำสั่ง (Mnemonic code) และส่วนที่เป็นตัวกระทำ (operand)

รหัสคำสั่ง คือ MOV เป็นคำสั่งให้ไมโครคอนโทรลเลอร์รู้หน้าที่การทำงานและการย้ายข้อมูลจากค่าตัวกระทำ คือ A,#data และนำเอาข้อมูลจากที่กำหนดโดยตรงมาเก็บไว้ในรีจีสเตอร์ A ที่อยู่ภายในไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณหาค่าเวลาในการทำงานของแต่ละคำสั่งของไมโครคอนโทรลเลอร์ทำให้สามารถทราบถึงเวลาในการทำงานของคำสั่งในโปรแกรมทั้งหมด เพื่อกำหนดและออกแบบวงจรควบคุมอุปกรณ์ภายนอกให้ทำงานสัมพันธ์กัน โดยเฉพาะค่าหน่วงเวลาการทำงานของอุปกรณ์ต่างๆ ซึ่งการทำงานในแต่ละคำสั่งจะใช้เวลาประมวลผลที่แตกต่างกัน ทุกคำสั่งจะมีค่าเวลาของรอบการทำงานที่แน่นอน และสามารถคำนวณหาค่าเวลาของรอบการทำงานได้จากวิธีการต่อไปนี้

$$\text{Time} = \text{Mc} \times \frac{12}{f_{crystal}} \qquad (2.1)$$

Time = เวลาที่ใช้ประมวลคำสั่งทั้งหมด

Mc = เวลารอบการทำงานของคำสั่ง

$F_{Crystal}$ = ค่าความถี่ของคริสตัลที่ต่อในวงจรกำเนิดสัญญาณให้กับไมโครคอนโทรลเลอร์ตระกูล MCS-51

2.2.6 ค่าตัวแปรที่กำหนดให้คำสั่ง

ในชุดคำสั่งของไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะใช้ค่าตัวแปรหรืออักษรย่อต่างๆ ในการกำหนดตำแหน่งการทำงาน ตารางรูปที่ 2.4

ตารางที่ 2.4 ค่าตัวแปรที่กำหนดในคำสั่ง

สัญลักษณ์	รายละเอียดการทำงานของตำแหน่ง
Rn	รีจีสเตอร์แบงก์ R0-R7 ในตำแหน่งที่กำหนดปัจจุบัน
direct	ข้อมูล 8 บิต เป็นตำแหน่งหน่วยความจำข้อมูล (RAM) เข้าถึงได้โดยตรง (0-127) หรือรีจีสเตอร์ SFR (128-256)
@Ri	ข้อมูล 8 บิต เป็นตำแหน่งหน่วยความจำข้อมูล (RAM) ภายใน เข้าถึงได้โดยตรง (0-256) โดยอ้างผ่านรีจีสเตอร์ R0 และ R1
#data	ค่าคงที่ขนาด 8 บิต เป็นการกำหนดข้อมูลโดยตรงในคำสั่ง
#data 16	ค่าคงที่ขนาด 16 บิต เป็นการกำหนดข้อมูลโดยตรงในคำสั่ง
Add 16	ข้อมูล 16 บิต เป็นค่าตำแหน่งการกระโดดไปของไมโครคอนโทรลเลอร์ในหน่วยความจำโปรแกรม ซึ่งสามารถกระโดดไปที่ระยะทาง 64 กิโลไบต์ ใช้กับคำสั่ง LCALL , LJMP
Add 11	ข้อมูล 11 บิต เป็นค่าตำแหน่งการกระโดดไปของไมโครคอนโทรลเลอร์ในหน่วยความจำโปรแกรม ซึ่งสามารถกระโดดไปที่ระยะทาง 64 กิโลไบต์ ใช้กับคำสั่ง ACALL , AJMP
rel	ข้อมูล 8 บิต เป็นการอ้างอิงระยะทางของไมโครคอนโทรลเลอร์ และจะกระโดดไปทำงานในระยะทางอ้างอิงตั้งแต่ -128 ถึง +127 ไบต์
bit	เป็นตำแหน่งบิตที่เข้าถึงได้โดยตรงของหน่วยความจำข้อมูลภายใน หรือรีจีสเตอร์SFR (special Function Register)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7 กลุ่มคำสั่งทางคณิตศาสตร์ (Arithmetic Instruction)

เป็นชุดคำสั่งที่ทำงานเกี่ยวกับการคำนวณทางคณิตศาสตร์ การบวก การลบ การคูณ การหาร การเพิ่มค่า และลดค่า ซึ่งจะใช้รีจีสเตอร์ Acc และ B เป็นตัวทำงานหลัก โดยคำสั่งการลดค่าและเพิ่มค่ากระทำได้กับรีจีสเตอร์ทั่วไปและหน่วยความจำ ได้แสดงดังตารางที่ 2.5

ตารางที่ 2.5 กลุ่มคำสั่งทางคณิตศาสตร์

รหัสคำสั่ง	รายละเอียดของคำสั่ง	รหัสไบต์	จำนวนคาบเวลา
ADD A,Rn	บวกค่าในรีจีสเตอร์ Acc ด้วยรีจีสเตอร์ Rn	1	12
ADD A,direct	บวกค่าในรีจีสเตอร์ Acc ด้วยรีจิสเตอร์ direct	2	12
ADD A,@Ri	บวกค่าในรีจีสเตอร์ Acc ด้วย RAM ภายในผ่านรีจีสเตอร์ Ri	1	12
ADD A,#data	บวกค่าข้อมูลกับรีจีสเตอร์ Acc	2	12
ADDC A,Rn	บวกค่ารีจีสเตอร์ Acc ด้วยรีจีสเตอร์ Rn พร้อมบิตตัวทด	1	12
ADDC A,Direct	บวกค่ารีจีสเตอร์ Acc ด้วยค่าใน direct พร้อมบิตตัวทด	2	12
ADD A,@Ri	บวกค่าใน RAM ภายในกับรีจีสเตอร์ Acc พร้อมบิตตัวทด	1	12
ADDC A,#data	บวกค่าข้อมูลกับรีจีสเตอร์ Acc พร้อมบิตตัวทด	1	12
SUBB A,Rn	ลบค่าในรีจีสเตอร์ Acc ด้วยรีจีสเตอร์ Rn พร้อมบิตตัวทด	1	12
SUBB A,Direct	ลบค่าในรีจีสเตอร์ Acc ด้วยค่าใน direct พร้อมบิตตัวทด	1	12
SUBB A,@Ri	ลบค่าในรีจีสเตอร์ Acc ด้วยค่า RAM ภายใน พร้อมบิตตัวทด	2	12
SUBB A,#data	ลบค่าในรีจีสเตอร์ Acc ด้วยค่าข้อมูล พร้อมบิตตัวทด	1	12
INC A	เพิ่มค่าข้อมูลใน Acc ขึ้น 1 ค่า	1	12
INC Rn	เพิ่มค่าข้อมูลใน Rn ขึ้น 1 ค่า	2	12
INC Direct	เพิ่มค่าข้อมูลใน direct ขึ้น 1 ค่า	1	12
INC @Ri	เพิ่มค่าใน direct RAM ผ่านรีจีสเตอร์ Ri ขึ้น 1 ค่า	1	12
DEC A	ลดค่าในรีจีสเตอร์ Acc ลง 1 ค่า	1	12
DEC Rn	ลดค่าใน Rn ลง 1 ค่า	1	12
DEC @Ri	ลดค่าใน direct RAM ผ่านรีจีสเตอร์ Ri ลง 1 ค่า	1	12
INC DPTR	เพิ่มค่าในรีจีสเตอร์ DPTR ขึ้น 1 ค่า	1	24
MUL AB	คูณค่าในรีจีสเตอร์ Acc ด้วยรีจีสเตอร์ B	2	48
DLV AB	หารค่ารีจีสเตอร์ Acc ด้วยรีจีสเตอร์ B	2	48
DA A	ปรับค่าในรีจีสเตอร์ Acc ให้เป็นฐาน 10	1	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.8 กลุ่มคำสั่งการกระทำลอจิก (Logical Instruction)

กลุ่มคำสั่งที่ทำงานเกี่ยวกับการกระทำทางลอจิกเบื้องต้น ได้แก่ วงจรลอจิก AND, OR, NOT, X-OR การหมุนและเลื่อนค่าในรีจีสเตอร์ การแลกเปลี่ยนของค่าระหว่างรีจีสเตอร์และหน่วยความจำ ดังตารางที่ 2.6

ตารางที่ 2.6 กลุ่มคำสั่งการกระทำลอจิก

รหัสคำสั่ง	รายละเอียดของคำสั่ง	รหัสไบต์	จำนวนคาบเวลา
ANL A,Rn	AND ค่ารีจีสเตอร์ Acc ด้วยรีจีสเตอร์ Rn	1	12
ANL A.direct	AND ค่ารีจีสเตอร์ Acc ด้วยด้วยค่าใน direct	2	12
ANL A,@Ri	AND ค่ารีจีสเตอร์ Acc ด้วยค่าในหน่วยความจำผ่านค่ารีจีสเตอร์Ri (R0,R1)	1	12
ANL A,#data	AND ค่ารีจีสเตอร์ Acc ด้วยค่าข้อมูลโดยตรง	2	12
ANL direct,A	AND ค่าใน direct ด้วยค่าในรีจีสเตอร์ Acc	2	12
ANL direct,#data	AND ค่าใน direct ด้วยค่าข้อมูลโดยตรง	3	12
ORL A,Rn	ORL ค่ารีจีสเตอร์ Acc ด้วยรีจีสเตอร์ Rn	1	12
ORL A,direct	ORL ค่ารีจีสเตอร์ Acc ด้วยด้วยค่าใน direct	2	12
ORL A,@Ri	ORL ค่ารีจีสเตอร์ Acc ด้วยค่าในหน่วยความจำผ่านค่ารีจีสเตอร์ Ri (R0,R1)	1	12
ORL A,#data	ORL ค่ารีจีสเตอร์ Acc ด้วยค่าข้อมูลโดยตรง	2	12
ORL direct,A	ORL ค่าใน direct ด้วยค่าในรีจีสเตอร์ Acc	2	12
ORL direct,#data	ORL ค่าใน direct ด้วยค่าข้อมูลโดยตรง	3	12
XRL A,Rn	XRL ค่ารีจีสเตอร์ Acc ด้วยรีจีสเตอร์ Rn	1	12
XRL A,direct	XRL ค่ารีจีสเตอร์ Acc ด้วยด้วยค่าใน direct	2	12
XRL A,@Ri	XRL ค่ารีจีสเตอร์ Acc ด้วยค่าในหน่วยความจำผ่านค่ารีจีสเตอร์ Ri (R0,R1)	1	12
XRL A,#data	XRL ค่ารีจีสเตอร์ Acc ด้วยค่าข้อมูลโดยตรง	2	12
XRL direct,A	XRL ค่าใน direct ด้วยค่าในรีจีสเตอร์ Acc	2	12
XRL direct,#data	XRL ค่าใน direct ด้วยค่าข้อมูลโดยตรง	3	24
CLR A	ทำให้ค่ารีจีสเตอร์ Acc มีค่าเป็น 0 ทุกบิต	1	12
CPL A	กลุ่มค่าในรีจีสเตอร์ Acc ให้มีค่าจาก 0 → 1 และจาก 1 → 0	1	12
RL A	หมุนค่าในรีจีสเตอร์ Acc จาก 7 ไปบิต 0	1	12
RLC A	หมุนค่าในรีจีสเตอร์ Acc จาก 7 ผ่านบิตตัวทดไปบิต 0	1	12
RR A	หมุนค่าในรีจีสเตอร์ Acc จาก 0 ไปบิต 7	1	12
RRC A	หมุนค่าในรีจีสเตอร์ Acc จาก 0 ผ่านบิตตัวทดไปบิต 7	1	12
SWAP A	สลับค่าบิตภายในรีจีสเตอร์ Acc จาก $(A_{3-0}) \leftrightarrow (A_{7-4})$	1	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
62850

2.2.9 กลุ่มคำสั่งการเคลื่อนย้ายข้อมูล (Data Transfer Instruction)

คำสั่งการเคลื่อนย้ายข้อมูลเป็นการคัดลอกข้อมูลระหว่างรีจีสเตอร์กับรีจีสเตอร์ รีจีสเตอร์กับหน่วยความจำทั้งภายในและภายนอก และหน่วยความจำกับหน่วยความจำ แสดงไว้ดังตารางที่ 2.7

ตารางที่ 2.7 กลุ่มคำสั่งการเคลื่อนย้ายข้อมูล

รหัสคำสั่ง	รายละเอียดของคำสั่ง	รหัสไบต์	จำนวนคาบเวลา
MOV A,Rn	คัดลอกข้อมูลจากรีจีสเตอร์ Rn ไปที่รีจีสเตอร์ Acc	1	12
MOV A,direct	คัดลอกข้อมูลจาก direct ไปที่รีจีสเตอร์ Acc	2	12
MOV A.@Ri	คัดลอกข้อมูลจากหน่วยความจำ RAM ไปที่รีจีสเตอร์ Acc	1	12
MOV A,#data	คัดลอกข้อมูลโดยตรงไปที่รีจีสเตอร์ Acc	2	12
MOV Rn,A	คัดลอกข้อมูลจากรีจีสเตอร์ Acc ไปที่รีจีสเตอร์ Rn	1	12
MOV Rn,direct	คัดลอกข้อมูลจาก direct ไปที่รีจีสเตอร์ Rn	2	24
MOV Rn,#data	คัดลอกข้อมูลจากข้อมูลโดยตรงไปที่รีจีสเตอร์ Rn	2	12
MOV direct,A	คัดลอกข้อมูลจากรีจีสเตอร์ Acc ไปที่ direct	2	12
MOV direct,Rn	คัดลอกข้อมูลจากรีจีสเตอร์ Rn ไปที่ direct	2	24
MOV direct,direct	คัดลอกข้อมูลจาก direct ไปที่ direct	3	24
MOV direct.@Ri	คัดลอกข้อมูลจากหน่วยความจำ RAM ไปที่ direct	2	24
MOV direct,#data	คัดลอกข้อมูลจากข้อมูลโดยตรงไปที่ direct	3	24
MOV @Ri,A	คัดลอกข้อมูลจากรีจีสเตอร์ Acc ไปที่หน่วยความจำ RAM	1	12
MOV @Ri.direct	คัดลอกข้อมูลจาก direct ไปที่หน่วยความจำ RAM	2	24
MOV @Ri,#data	คัดลอกข้อมูลจากข้อมูลไปที่หน่วยความจำ RAM	2	24
MOV DPTR,#DATA 16	คัดลอกข้อมูล 16 บิต จากข้อมูลไปที่รีจีสเตอร์ DPTR	3	24
MOVC A,@A+DPTR	คัดลอกข้อมูลจากตำแหน่ง A+DPTR ไปที่รีจีสเตอร์ Acc	1	24
MOVC A,@A+PC	คัดลอกข้อมูลจากตำแหน่ง A+PC ไปที่รีจีสเตอร์ Acc	1	24
MOVX @Ri,A	คัดลอกข้อมูลจาก RAM ภายนอกไปที่รีจีสเตอร์ Acc	1	24
MOVX @DPTR,A	คัดลอกข้อมูลจากรีจีสเตอร์ Acc ไปที่รีจีสเตอร์ DPTR กำหนด	1	24
PUSH DIRECT	นำข้อมูล 8 บิต ไปเก็บในสแตก (SP)	2	24
POP DIRECT	นำข้อมูล 8 บิต ออกเก็บในสแตก (SP)	2	24
XCH A,Rn	แลกเปลี่ยนข้อมูลระหว่างรีจีสเตอร์ Acc กับรีจีสเตอร์ Rn	1	12
XCH A,DIRECT	แลกเปลี่ยนข้อมูลระหว่างรีจีสเตอร์ Acc กับหน่วยความจำ RAM	2	12
XCH A,@Ri	แลกเปลี่ยนข้อมูลระหว่างรีจีสเตอร์ Acc กับรีจีสเตอร์ Ri กำหนดตำแหน่ง	1	12
XCHD A,@Ri	แลกเปลี่ยนบิต 3-0 ระหว่างรีจีสเตอร์ Acc กับตำแหน่ง direct	1	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.10 กลุ่มคำสั่งการจัดการข้อมูลระดับบิต (Bit Manipulated Instruction)

เป็นกลุ่มของคำสั่งที่ทำหน้าที่เกี่ยวกับหน่วยความจำที่เข้าถึงระดับบิตโดยตรง โดยการเปลี่ยนแปลงค่าข้อมูลระดับบิต การทำให้มีค่าสภาวะลอจิกเป็น "0" หรือ "1" การกลับค่าบิต แสดงดังตารางที่ 2.8

ตารางที่ 2.8 กลุ่มคำสั่งการจัดการข้อมูลระดับบิต

รหัสคำสั่ง	รายละเอียดของคำสั่ง	รหัสไบต์	จำนวนคาบเวลา
CLR C	ทำให้บิตตัวทดมีค่าเป็น 0	1	12
CLR bit	ทำให้บิตที่กำหนดมีค่าเป็น 0	2	12
SETB C	กำหนดให้บิตตัวทดมีค่าเป็น 1	1	12
SETB bit	ทำให้บิตที่กำหนดมีค่าเป็น 1	2	12
CPL	กลับค่าในบิตตัวทดจาก 0 → 1, 1 → 0	1	12
CPL bit	กลับค่าในบิตที่กำหนดตัวทดจาก 0 → 1, 1 → 0	2	24
ANL C,bit	AND บิตตัวทดด้วยบิตที่กำหนด	2	24
ANL C,/bit	AND บิตตัวทดด้วยบิตที่กำหนดกลับค่า	2	12
ORL C,bit	OR บิตตัวทดด้วยบิตที่กำหนด	2	24
ORL C,/bit	OR บิตตัวทดด้วยบิตที่กำหนดกลับค่า	2	24
MOV C,bit	คัดลอกบิตที่กำหนดไปบิตตัวทด	2	24
MOV bitj,C	คัดลอกบิตตัวทดไปบิตที่กำหนด	2	24
JC rel	กระโดดไปที่ตำแหน่ง rel เมื่อบิตตัวทดมีค่าเป็น 1	2	24
JNC rel	กระโดดไปที่ตำแหน่ง rel เมื่อบิตตัวทดมีค่าไม่เป็น 1	2	24
JB bit,rel	กระโดดไปเมื่อบิตที่กำหนดมีค่าเป็น 1	3	24
JNB bit,rel	กระโดดไปเมื่อบิตที่กำหนดมีค่าไม่เป็น 1	3	24
JBC bit,rel	กระโดดไปเมื่อบิตกำหนดเป็น 1 และทำให้บิตนั้นเป็น 0	3	24

2.2.11 กลุ่มคำสั่งการกระโดด (Branch Instruction)

เป็นกลุ่มคำสั่งที่ทำหน้าที่เกี่ยวกับการกระโดดไปที่ตำแหน่งที่โปรแกรมกำหนด โดยมีลักษณะการกระโดดอยู่ 2 แบบ คือ การกระโดดโดยไม่มีเงื่อนไข เมื่อพบคำสั่งก็กระโดดไปตำแหน่งที่กำหนดทันทีและการกระโดดโดยมีเงื่อนไข เมื่อไมโครคอนโทรลเลอร์พบคำสั่งจะต้องตรวจสอบเงื่อนไขก่อนว่าเป็นไปตามที่กำหนดหรือไม่ ถ้าถูกต้องจึงกระโดดไปที่ตำแหน่งที่กำหนด

การกระโดดไปทำงานก็มีรูปแบบคือ กระโดดไปโดยเป็นการเรียกใช้โปรแกรมย่อย โดยจะต้องกลับมาทำงานได้ก็ต่อเมื่อเสร็จจากการทำงานในโปรแกรมย่อย (ACALL, LCALL) และกระโดดไปทำงานโดยไม่กลับมา (SJMP, LJMP) ระยะทางการกระโดดก็มีให้เลือกเป็นระยะแบบสั้น (256 ไบต์ หรือ 2 กิโลไบต์) ระยะ (64 กิโลไบต์) ดังแสดงไว้ในตารางที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.9 กลุ่มคำสั่งการกระโดด

รหัสคำสั่ง	รายละเอียดของคำสั่ง	รหัสไบต์	จำนวนคาบเวลา
ACALL add11	เรียกโปรแกรมย่อยที่ addr11	2	24
LCALL addr16	เรียกโปรแกรมย่อยที่ addr16	3	24
RET	กลับจากโปรแกรมย่อยไปโปรแกรมหลัก	1	24
RETI	กลับจากการเรียกใช้อินเตอร์รัปต์	1	24
AJMP addr11	กระโดดทำงานที่กำหนดตำแหน่งด้วย addr11	2	24
LJMP addr16	กระโดดทำงานที่กำหนดตำแหน่งด้วย addr16	3	24
SJMP rel	กระโดดไปทำงานที่กำหนดตำแหน่งด้วย rel	2	24
JMP @A+DPTR	กระโดดไปทำงานที่ตำแหน่ง A+DPTR	1	24
JZ rel	กระโดดไปทำงานที่ rel เมื่อค่าในรีจีสเตอร์ Acc = 00	2	24
JNZ rel	กระโดดไปทำงานที่ rel เมื่อค่าในรีจีสเตอร์ Acc ≠ 00	2	24
CJNE A,direct,rel	ถ้า direct ≠ Acc กระโดดไปที่ rel	3	24
CJNE A,#data,rel	ถ้า #data ≠ Acc กระโดดไปที่ rel	3	24
CJNE Rn,#data,rel	ถ้า direct ≠ Rn กระโดดไปที่ rel	3	24
CJNE @Ri,#data,rel	ถ้า direct ≠ Ri กระโดดไปที่ rel	3	24
DJNZ Rn,rel	ถ้า Rn -1 ≠ 0 กระโดดไปที่ rel	2	24
DJNZ direct,rel	ถ้า direct – 1 ≠ 0 กระโดดไปที่ rel	3	24
NOP	เพิ่มค่า PC ขึ้น 1 ค่า (ไม่มีการทำงานอย่างอื่น)	1	12

2.2 การเชื่อมต่อแบบไร้สาย (Wireless RF Module)

ทุกวันนี้อะไรๆ ก็ดูพัฒนาให้เป็นอุปกรณ์ไร้สายไปเสียหมด เหตุสำคัญก็เพราะอุปกรณ์บางอย่างนั้นเมื่อพัฒนาให้เป็นอุปกรณ์ไร้สายแล้วจะทำให้มันมีความสะดวกใช้มากขึ้น และยังได้รับการยอมรับจากผู้ใช้อย่างรวดเร็ว ตัวอย่างที่เห็นได้ชัดก็คือโทรศัพท์มือถือ หรืออุปกรณ์ป้องกันภัยอย่างรีโมท ในปัจจุบัน เทคโนโลยีของการเชื่อมต่อแบบไร้สายนั้นมีอยู่หลากหลายแบบ แต่ส่วนใหญ่แล้วก็ใช้เทคโนโลยีการเชื่อมต่อด้วยแสงหรือไม่ก็ใช้คลื่นวิทยุ ซึ่งต่างก็มีข้อดีข้อด้อยแตกต่างกันไป ในโครงงานนี้จะเป็นตัวอย่างของการประยุกต์ใช้ในรูปแบบหนึ่งเท่านั้น โดยโครงงานนี้จะมีสาระสำคัญอยู่ที่การนำตัว Wireless RF Module มาใช้ทำหน้าที่ส่งข้อมูล

- **หลักการทำงาน**

ชุด RF-Module Wireless 433 MHz คือ วงจรภาคส่ง และวงจรภาครับ ซึ่งโมดูลนี้จะเป็นแบบรับ-ส่งทางเดียว (Transmitter and Receiver) ใช้การโมดูเลชั่นแบบ ASK Modulation โดยหลักการคือเมื่อมีการป้อนแรงดันไปยังภาคส่ง ภาคส่งจะทำการส่งข้อมูลไปยังภาครับ โดยเทคนิคของการส่งข้อมูลของชุด RF (Transmitter) Module นี้ ในขั้นแรกจำเป็นจะต้องมีการส่งรหัสนำหน้าหรือที่เรียกว่า Pre amble ในลักษณะ 010101010101 ออกไปก่อนประมาณ 2 ไบต์ และหลังจากส่งค่า Pre amble แล้วให้ส่ง Code Check โดยใช้ไบต์ข้อมูล j และ U เพื่อให้ภาครับทำการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นก็จะทำการส่งค่ารหัสหมายเลขตามไปอีก 1 ไบต์ข้อมูล สำหรับข้อมูลนี้ไม่จำเป็นว่าจะต้องมีไบต์เดียวเสมอไป เพียงแต่ขั้นตอนในการส่งนั้นจะต้องมีการส่งค่า Pre amble และ Code Check ก่อนเสมอแล้วตามด้วยข้อมูลหลายๆ ไบต์ก็ได้ และหากจะต้องมีการส่งวนซ้ำให้หน่วงเวลา (Delay) ไว้ประมาณ 2 วินาที

ส่วนเทคนิคของการรับข้อมูลนั้นจะเป็นเทคนิคของการรับข้อมูลของชุดรับข้อมูล RF (Receiver) Module ไม่ต้องทำการตรวจสอบการรับ Pre amble ให้คอยตรวจสอบการรับค่า j และ U หากรับข้อมูลที่เป็นค่า j และไบต์ข้อมูลตัวต่อไปคือ U หากรับได้ค่าตามนี้ไบต์ข้อมูลถัดมาจะเป็นค่าของไบต์ข้อมูลจริงที่ต้องการส่งมาจากภาคส่ง

2.4 โปรแกรมแอสเซมบลี

ภาษาที่ใช้รหัสนีโมนิคในการเขียนโปรแกรมเรียกว่า ภาษาแอสเซมบลี (Assembly language) ซึ่งเป็นภาษาระดับต่ำ แต่เครื่องคอมพิวเตอร์ไม่สามารถเข้าใจรหัสนี้ได้ จึงต้องมีการแปลงภาษาแอสเซมบลีให้เป็นภาษาที่เครื่องเข้าใจโดยใช้ตัวแปลภาษาที่เรียกว่า แอสเซมเบลอร์ (Assembler) โดยรหัสนีโมนิคที่ใช้จะขึ้นกับ CPU แต่ละตระกูลด้วย ในการเขียนโปรแกรมภาษาแอสเซมบลี ผู้เขียนจะต้องเข้าใจสถาปัตยกรรมของไมโครโปรเซสเซอร์ที่ใช้กับคอมพิวเตอร์รุ่นนั้น ๆ เข้าใจการใช้งานรีจีสเตอร์แต่ละตัวว่ามีหน้าที่อย่างไร และเมื่อคอมพิวเตอร์ทำงานแต่ละคำสั่งเสร็จ จะมีผลต่อรีจีสเตอร์ใดบ้าง ซึ่งได้กล่าวไปในเรื่องของไมโครคอนโทรลเลอร์แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและวิธีการดำเนินงาน

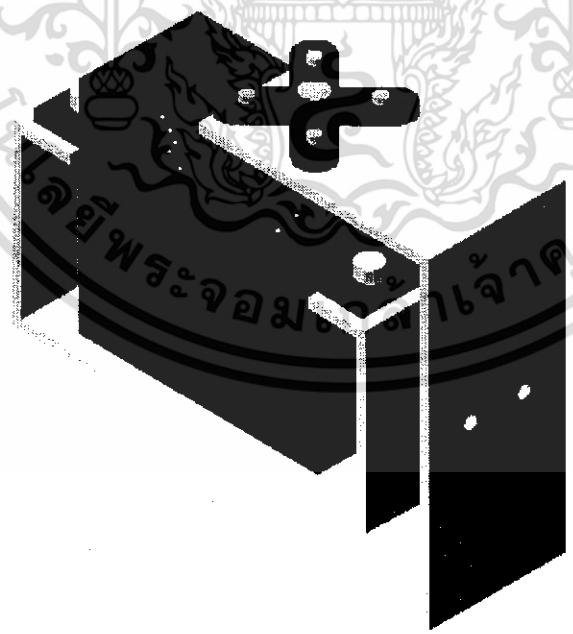
ในส่วนของการออกแบบและวิธีการดำเนินงานสามารถแบ่งรายละเอียดเป็นขั้นตอนได้ดังต่อไปนี้

3.1 ศึกษารวบรวมข้อมูล

ในขั้นตอนการดำเนินงานนี้ได้ทำการศึกษาข้อมูลที่มีความจำเป็นในการสร้างหุ่นยนต์ที่มีเซอร์โวมอเตอร์เป็นส่วนของข้อต่อ และตัดสินใจในการเลือกใช้วัสดุที่นำมาทำเป็นโครงสร้างของหุ่นยนต์ หลังจากที่ได้ศึกษาแล้วลงมือทำในขั้นตอนต่อไป

3.2 ออกแบบและเขียนแบบหุ่นยนต์

ในขั้นตอนนี้จะเป็นการออกแบบส่วนต่างๆ ของหุ่นยนต์ ไม่ว่าจะเป็นส่วนลำตัว แขน ขา ซึ่งเมื่อวิเคราะห์แล้วจึงได้ข้อสรุปว่าควรออกแบบส่วนของข้อต่อหุ่นยนต์ ให้เป็นส่วนประกอบที่เหมือนๆ กัน โดยมีลักษณะเป็นท่อนๆ เหมือนกับท่อนแขนหรือท่อนขาของมนุษย์ ดังแสดงไว้ในรูป 3.1 เพื่อให้ง่ายต่อการปรับปรุงแก้ไขหุ่นยนต์ ถ้าเกิดปัญหา และงานเขียนแบบทั้งหมดได้แสดงไว้ในภาคผนวก ก

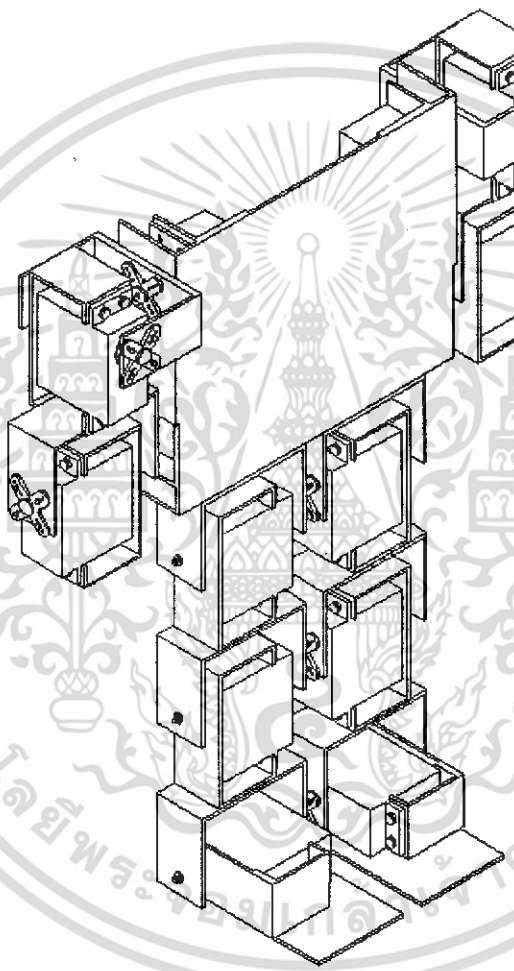


รูปที่ 3.1 แผนภาพการออกแบบข้อต่อที่เลียนแบบท่อนแขนหรือท่อนขาของมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ดำเนินการสร้างหุ่นยนต์

ในขั้นตอนนี้จะเป็นการนำรูปที่ได้ออกแบบไว้มาเป็นแบบในการสร้างส่วนต่างๆ ของหุ่นยนต์ และทำการประกอบส่วนต่างๆ ลำตัว แขนและขาเข้าด้วยกัน โดยหุ่นยนต์ตัวนี้ ได้ใช้เซอร์โวมอเตอร์ทั้งหมด 14 ตัว คือในส่วนของขานั้นเราจะใช้เซอร์โวมอเตอร์ 3 ตัว ด้วยกันซึ่งเลียนแบบข้อเท้า หัวเข่า และต้นขาของมนุษย์ ในส่วนของแขนนั้นเราใช้ เซอร์โวมอเตอร์เพียง 2 ตัว คือ ส่วนที่เป็นข้อศอกและหัวไหล่ ในส่วนของลำตัวนั้น จะใช้เซอร์โวมอเตอร์ 4 ตัว คือ สองตัวบน ทำให้หัวไหล่สามารถเลียนแบบท่าทางการยกแขนได้ และสองตัวล่าง ทำให้ในส่วนของต้นขานั้นสามารถหมุนได้ ซึ่งส่วนประกอบต่างๆ ได้แสดงไว้ดังรูปที่ 3.2

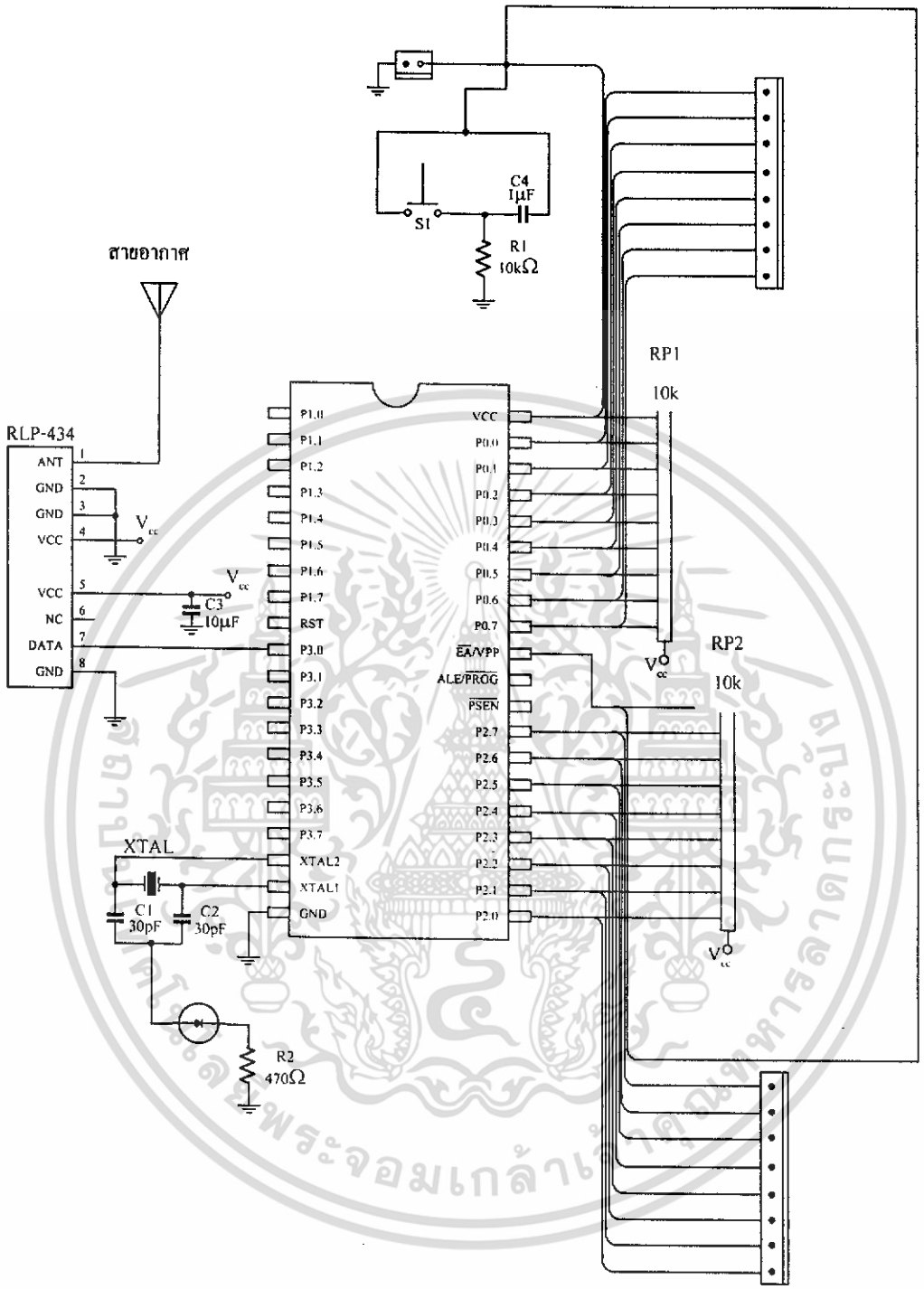


รูปที่ 3.2 แผนภาพแสดงส่วนประกอบต่างๆ ของหุ่นยนต์

3.4 สร้างแผงวงจรไมโครคอนโทรลเลอร์, วงจรรับ-ส่งสัญญาณระบบไร้สาย

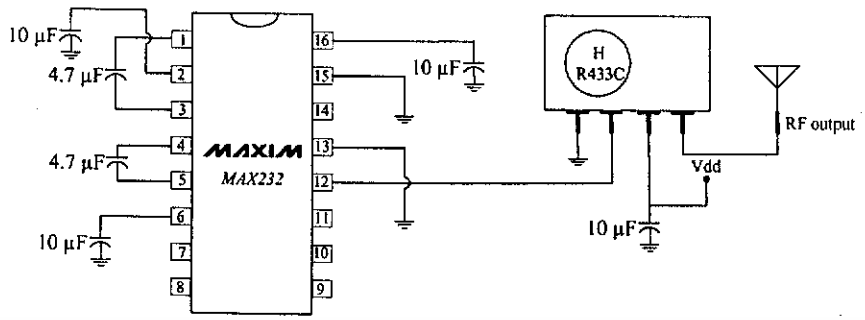
ในขั้นตอนจะเป็นสร้างแผงวงจรที่ไว้ใช้เป็นส่วนในการสั่งการให้หุ่นยนต์สามารถเคลื่อนที่ได้ แผงวงจรไมโครคอนโทรลเลอร์และวงจรรับสัญญาณระบบไร้สายแสดงไว้ดังรูปที่ 3.3 และวงจรส่งสัญญาณระบบไร้สายแสดงไว้ดังรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แผงวงจรไมโครคอนโทรลเลอร์และแผงวงจรรับสัญญาณระบบไร้สาย

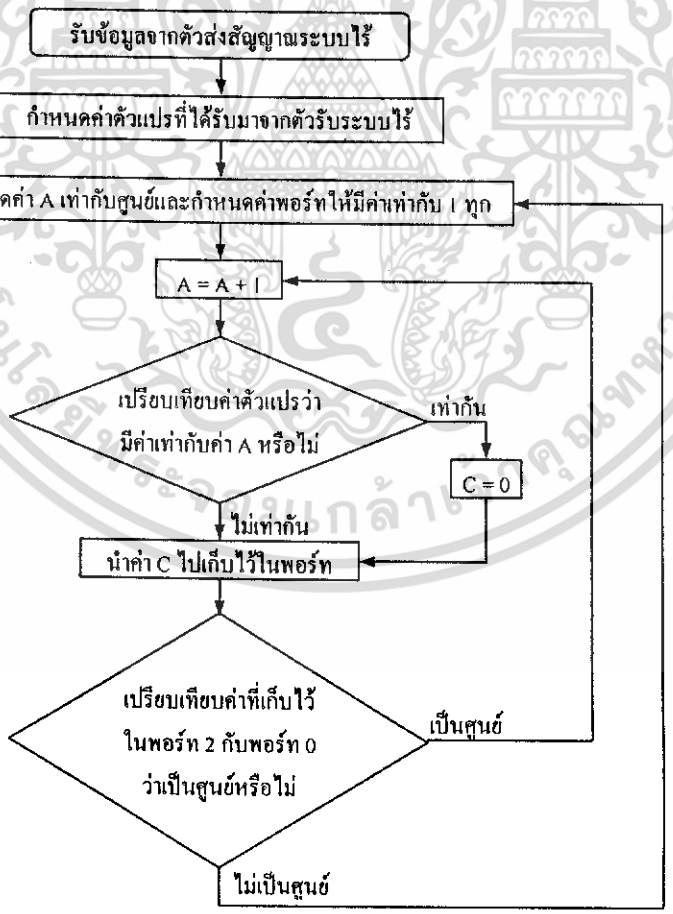
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แผงวงจรส่งสัญญาณระบบไร้สาย

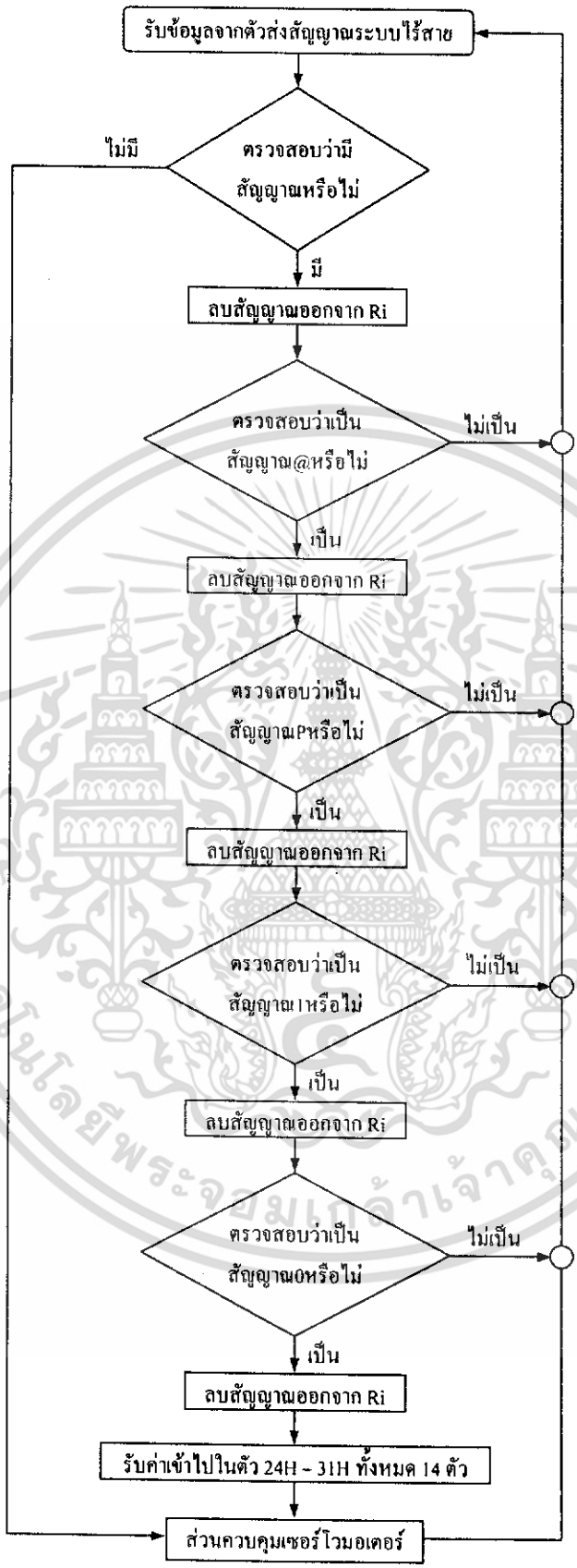
3.5 ออกแบบโปรแกรมสั่งการเซอร์โวมอเตอร์

ในขั้นตอนนี้จะเป็นออกแบบการสั่งการให้เซอร์โวมอเตอร์ทำงาน จากทฤษฎีของเซอร์โวมอเตอร์นั้นจะเห็นได้ว่าการทำงานของเซอร์โวมอเตอร์นั้น ค่าความกว้างพัลส์จะเป็นตัวกำหนดองศาของเซอร์โวมอเตอร์ และจะต้องมีการจ่ายสัญญาณพัลส์ให้กับเซอร์โวมอเตอร์ทุกตัวในทุกๆ 20 มิลลิวินาที เพื่อรักษาสภาพตำแหน่งของเซอร์โวมอเตอร์ไว้ จึงทำให้กลุ่มผู้วิจัยออกแบบโปรแกรมภาษาแอสเซมบลีสั่งการเซอร์โวมอเตอร์ ดังรูปที่ 3.5 รูปที่ 3.6



รูปที่ 3.5 แผนผังแสดงการออกแบบโครงสร้างโปรแกรมในส่วนของโปรแกรมภาษาแอสเซมบลี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แผนผังแสดงการออกแบบโปรแกรมในส่วนของระบบไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ขั้นตอนเขียนโปรแกรมและทดสอบโปรแกรม

ในขั้นตอนนี้จะเป็นการเขียนโปรแกรมทั้งสองส่วน คือ ส่วนของโปรแกรมภาษาแอสเซมบลี และในส่วนของระบบไร้สาย โดยเราจะเริ่มทำการเขียนในส่วนของโปรแกรมภาษาแอสเซมบลีก่อน โดยเมื่อทำการเขียนแล้วก็นำมาทำการเช็คสัญญาณที่ขาของไมโครคอนโทรลเลอร์ว่ามีสัญญาณ และสามารถสั่งการให้เซอร์โวมอเตอร์ทำงานได้หรือไม่ เมื่อทดสอบโปรแกรมในส่วนนี้เสร็จเรียบร้อย เราก็จะทำการเริ่มเขียนโปรแกรมในส่วนของระบบไร้สาย ซึ่งจะใช้โปรแกรมวิชวลเบสิกเป็นตัวสั่งให้ตัวส่งสัญญาณระบบไร้สายส่งสัญญาณ

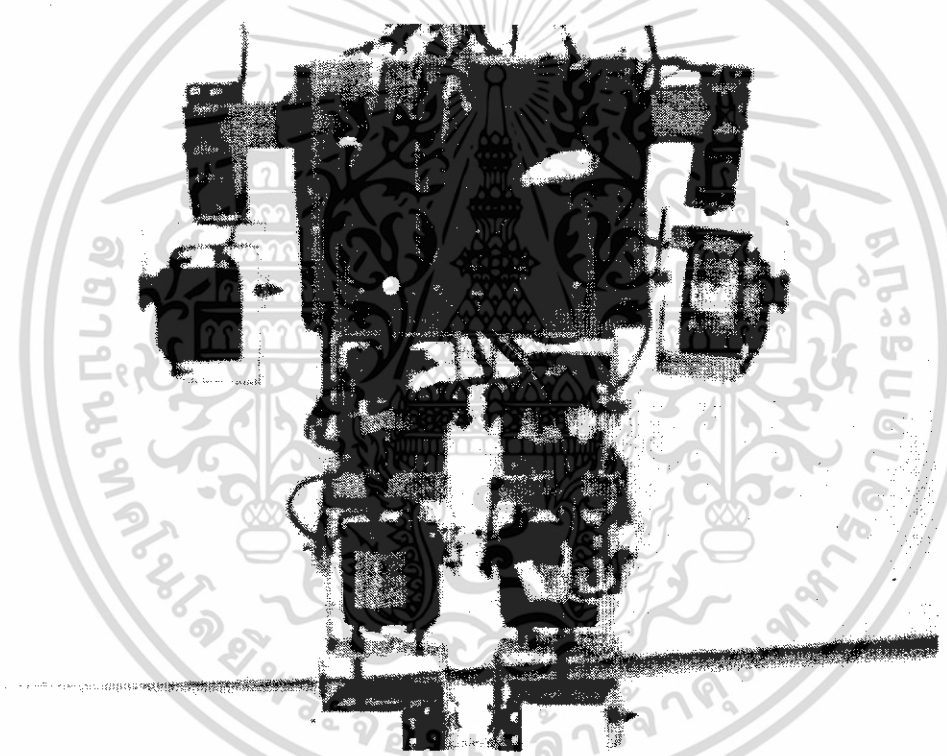
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการดำเนินงาน

4.1 ผลการดำเนินงานด้านฮาร์ดแวร์ – หุ่นยนต์ ROBO-ONE

ผลการดำเนินงานในด้านฮาร์ดแวร์นี้จะเป็นการสร้างหุ่นยนต์ คือการนำเซอร์โวมอเตอร์มาติดกับอลูมิเนียมแผ่นที่ได้ตัดและพับตามที่ได้ออกแบบไว้ มาประกอบเข้าด้วยกันกับส่วนตัว แขน ขา เท้า ตามรูปที่ 4.1

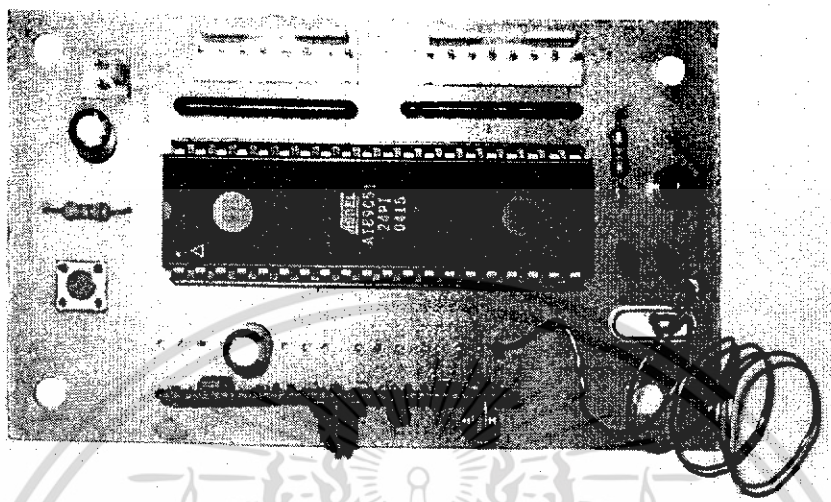


รูปที่ 4.1 รูปถ่ายแสดงผลการดำเนินงานด้านฮาร์ดแวร์

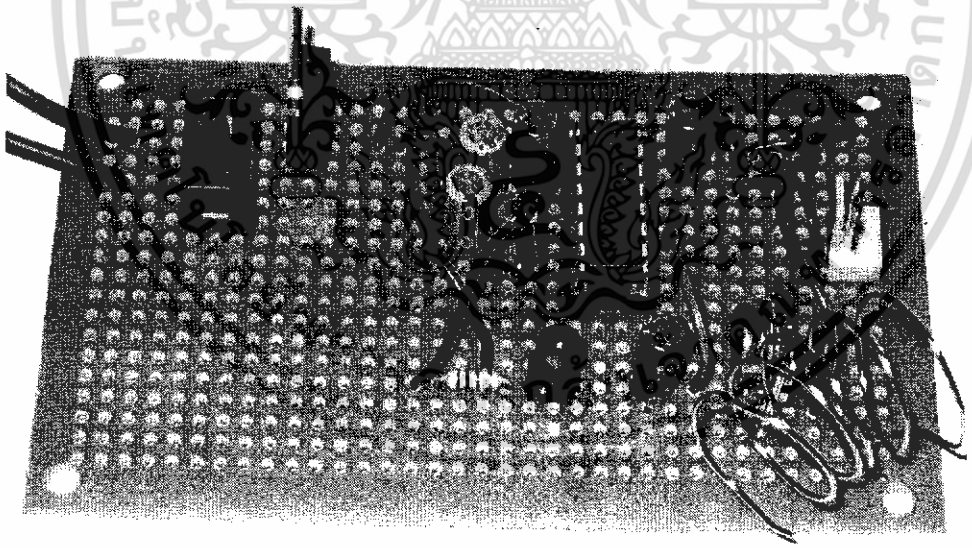
4.2 ผลการดำเนินงานทางด้านวงจรไฟฟ้า

ผลการดำเนินงานทางด้านวงจรไฟฟ้า คือ แผงวงจรไมโครคอนโทรลเลอร์และวงจรรับสัญญาณระบบไร้สายแสดงไว้ดังรูปที่ 4.2 และวงจรส่งสัญญาณระบบไร้สายแสดงไว้ดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 รูปถ่ายแผงวงจรไมโครคอนโทรลเลอร์และแผงวงจรรับสัญญาณระบบไร้สาย



รูปที่ 4.3 รูปถ่ายแผงวงจรส่งสัญญาณระบบไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบ

ผลการดำเนินงานของปริญญานิพนธ์ฉบับนี้ คือ หุ่นยนต์สามารถเคลื่อนไหวตามที่เราสั่งการได้จากโปรแกรมวิชวลเบสิก ว่าจะสั่งการให้เซอร์โวมอเตอร์ตัวไหนหมุนไปในทิศทางใด ตามที่เราต้องการได้ และเมื่อเราทดสอบในส่วนของระบบไร้สายนั้น พบว่าตัวส่งสัญญาณไร้สายสามารถส่งสัญญาณได้ระยะทางประมาณ 3 เมตร ซึ่งในเบื้องต้นหุ่นยนต์สามารถยืน และเลียนแบบท่าทางเดินของมนุษย์ได้แล้ว แต่ยังมีปัญหาที่ควบคุมลักษณะท่าทางการเดินได้ไม่ค่อยเหมือนมนุษย์ โดยที่กลุ่มผู้พัฒนาได้ลองหลายวิธีแล้ว และอีกทั้งยังเวลาค่อนข้างน้อยเกินไปจึงทำให้หุ่นยนต์ยังไม่สามารถเดินได้จริงในตอนนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิเคราะห์ผลการดำเนินงาน

5.1 รายละเอียดของปริญญานิพนธ์

โครงงานปริญญานิพนธ์นี้เป็นการสร้างหุ่นยนต์แล้วสั่งการให้หุ่นยนต์สามารถเคลื่อนตามที่เรากำหนด อีกทั้งยังการสั่งการนั้นเป็นระบบไร้สาย

5.2 วัตถุประสงค์ของโครงงานปริญญานิพนธ์

- สร้างหุ่นยนต์เสร็จเรียบร้อยแล้ว
- สามารถเขียนโปรแกรมควบคุมเซอร์โวมอเตอร์เสร็จแล้ว

5.3 ประโยชน์ของโครงงานปริญญานิพนธ์

- ได้หุ่นยนต์ที่เคลื่อนไหวในท่าทางต่างๆ ตามที่เราต้องการได้
- สามารถนำความรู้ที่ได้จากการทำปริญญานิพนธ์ไปใช้ประโยชน์ด้านการแก้ไขโปรแกรมควบคุมที่มีลักษณะคล้ายๆกันในโครงงานปริญญานิพนธ์ฉบับนี้

5.4 ผลการดำเนินงาน

- โครงงานนี้สามารถทำตามวัตถุประสงค์ที่ได้บรรลุตามที่กำหนดไว้ในขอบเขตแล้ว ซึ่งสามารถโปรแกรมควบคุมเซอร์โวมอเตอร์ ได้หลายๆ ตัวพร้อมกัน โดยการใช้ไมโครคอนโทรลเลอร์ MCS51 ในการควบคุม และใช้คอมพิวเตอร์สั่งงานโดยผ่านระบบไร้สาย (RF Wireless module)

5.5 แนวทางในการปรับปรุงพัฒนาโครงงานปริญญานิพนธ์

เนื่องจากการที่ได้ลองทดสอบโปรแกรมพบว่าโปรแกรมของเราไม่สามารถที่จะรักษาสภาพของเซอร์โวมอเตอร์ให้คงอยู่ได้ จึงพบแนวทางในการปรับปรุงแบ่งได้เป็นสามส่วน คือ

5.5.1 การปรับปรุงในส่วนของวงจร

เนื่องจากไมโครคอนโทรลเลอร์ MCS-51 มีข้อจำกัดในการใช้ค่อนข้างสูง จึงอาจที่จะเปลี่ยนไมโครคอนโทรลเลอร์เป็นตัวใหม่ หรืออาจจะเปลี่ยนตัวรับส่งให้มีความถี่ของสัญญาณมากขึ้นกว่าเดิม ซึ่งการที่จะให้เซอร์โวมอเตอร์คงสภาพอยู่ได้นั้น จะต้องให้มีความถี่สัญญาณเท่ากับ 20 มิลลิวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.2 การปรับปรุงในส่วนของโปรแกรม

ในส่วนของโปรแกรมนั้นทางผู้วิจัยได้หาแนวทางมาหลายแนวทางในการเขียนโปรแกรมแล้ว แต่ยังไม่สามารถทำให้ตัวโปรแกรมนั้นมีขนาดเล็กพอที่จะให้สัญญาณที่เข้าเซอร์โวมอเตอร์มีความถี่มากขึ้น

5.5.3 การปรับปรุงในส่วนของโครงสร้างหุ่นยนต์

ในส่วนของโครงสร้างหุ่นยนต์นั้นอาจจะต้องออกแบบให้มีขนาดเล็กลง เพื่อที่จะทำให้น้ำหนักของหุ่นยนต์มีน้ำหนักน้อยลงทำให้เซอร์โวมอเตอร์ไม่ต้องออกแรงต้านกับแรงโน้มถ่วงโลกในการที่จะพยุงตัวของหุ่นยนต์ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

ผศ.ธีรวัฒน์ ประกอบผล, 2545. ระบบคอมพิวเตอร์และภาษาแอสแซมบลี. พิมพ์ครั้งที่ 4. กรุงเทพฯ : สำนักพิมพ์สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น).

วัชรินทร์ เคารพ , 2546. คู่มือการใช้งาน SERVO MOTOR. พิมพ์ครั้งที่ 1. บริษัท อีทีที จำกัด.

รศ.สมยศ จุณณะปิยะ, 2546. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51. พิมพ์ครั้งที่ 5. ภาควิชาวิศวกรรมคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

http://www.alldatasheet.com

http://www.datasheetcatalog.com

http://www.micro4dev.com

http://www.tpa.or.th

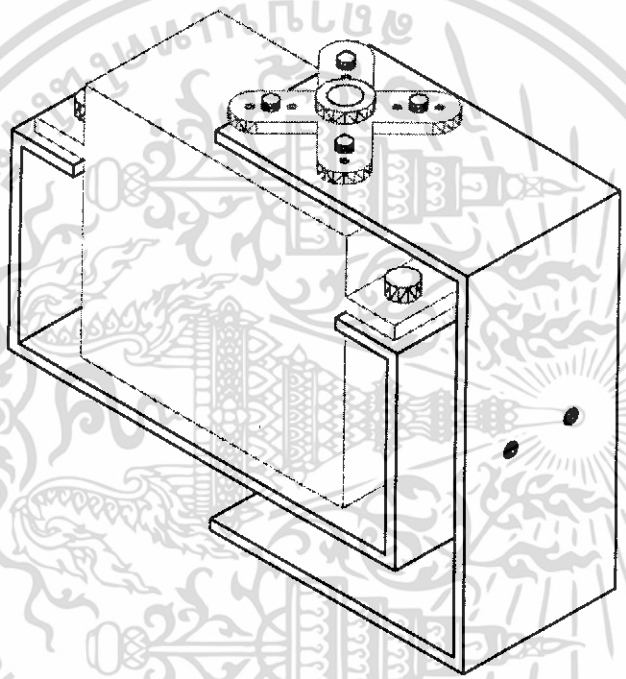
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ภาพการออกแบบของหุ่นยนต์ในปริญญานิพนธ์ฉบับนี้ จะมีดังต่อไปนี้

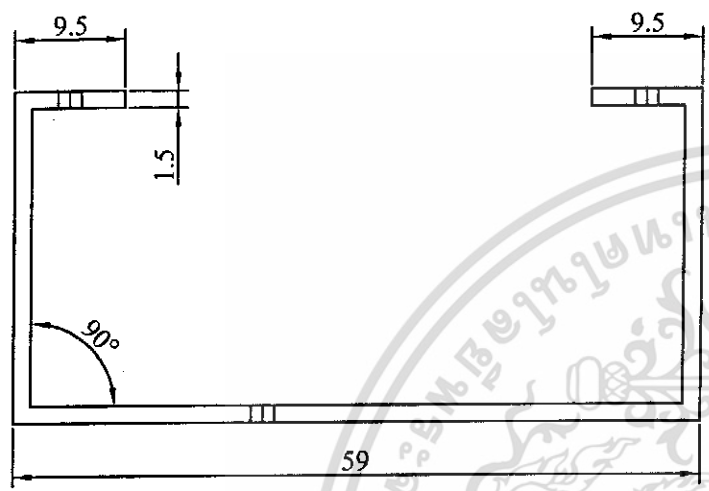
ROBOT		
Module	**Body**	**Foot**
-Part1	-Body	-Left Foot
-Part2		-Right Foot
-Part3		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

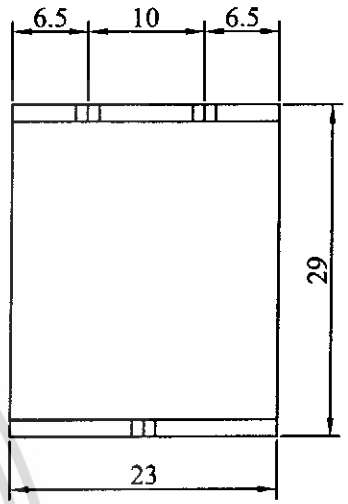


NO. of Piece	Nomenclature : Module	Scale 1 : 1
1/11	List of Parts : Module	3D View
King Mongkut's Institute of Tecnology		Date : 16/10/2005

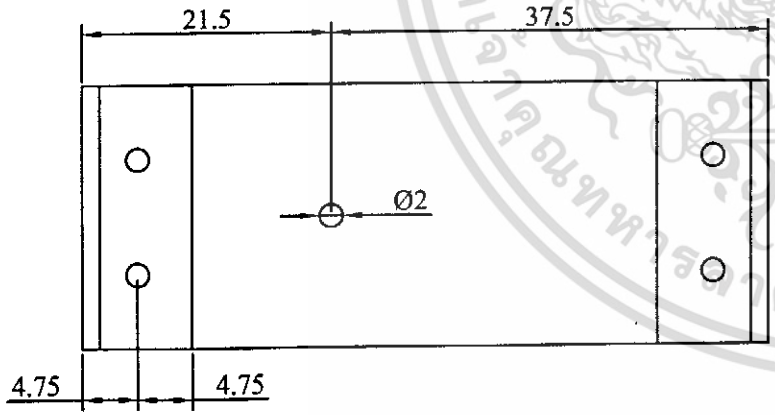
FIG 1



FRONT VIEW



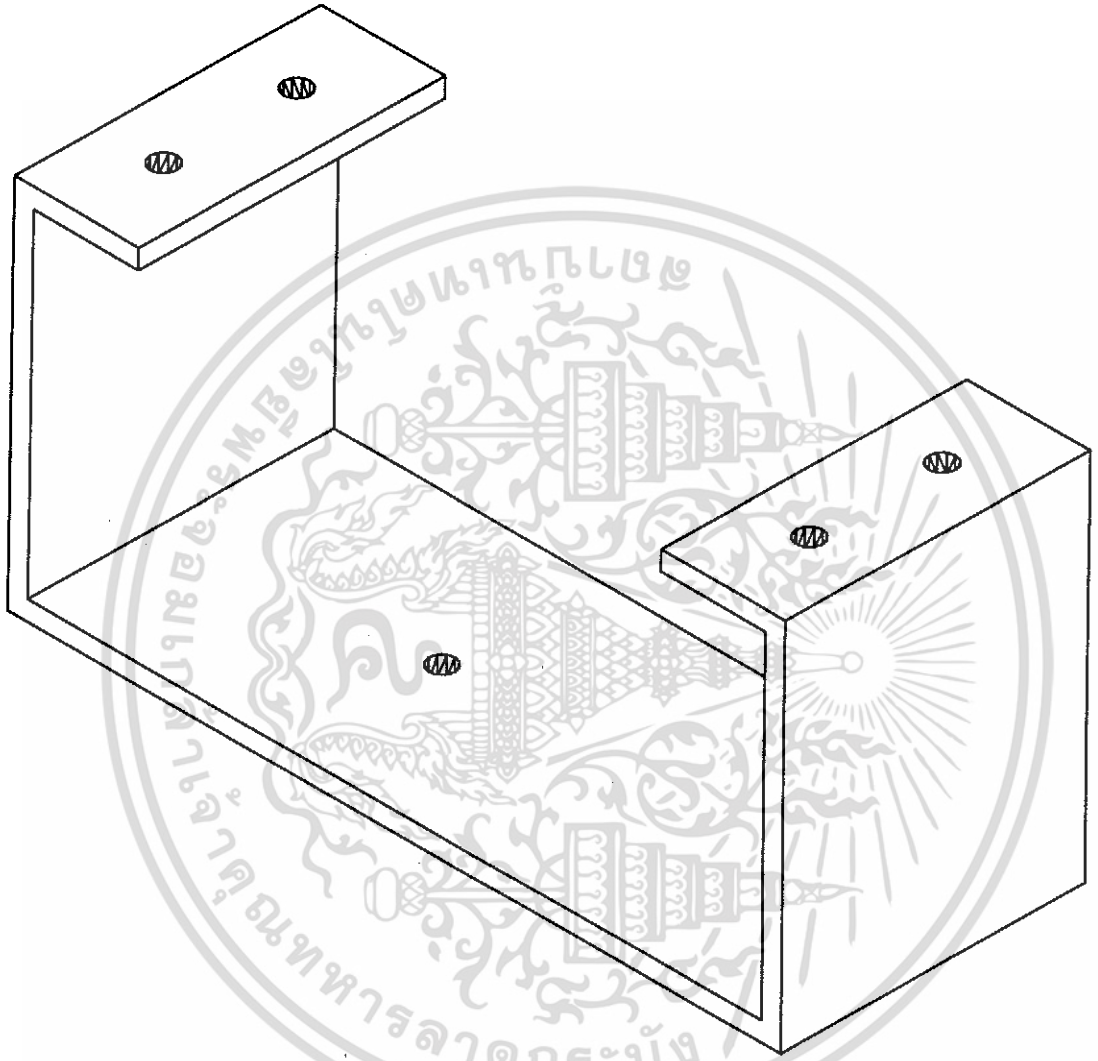
SIDE VIEW



TOP VIEW

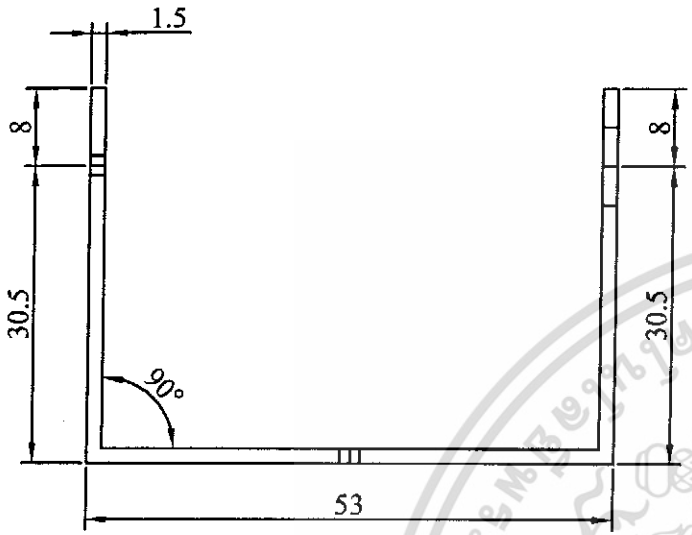
NO. of Piece	Nomenclature : Module	Scale 1 : 1
2/11	List of Parts : Part1	Top View/ Front View/ Side View
King Mongkut's Institute of Tecnology		Date : 16/10/2005

MM 2

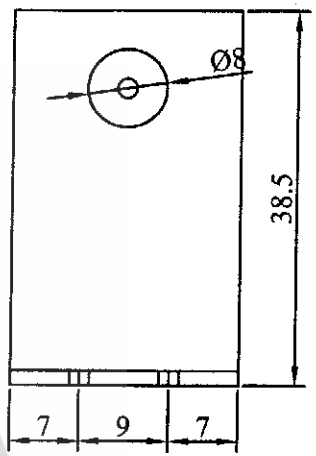


NO. of Piece	Nomenclature : Module	Scale 4 : 1
3/11	List of Parts : Part1	3D View
King Mongkut's Institute of Tecnology		Date : 16/10/2005

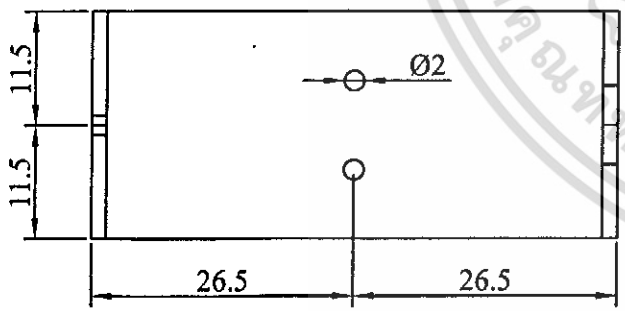
หก 3



FRONT VIEW



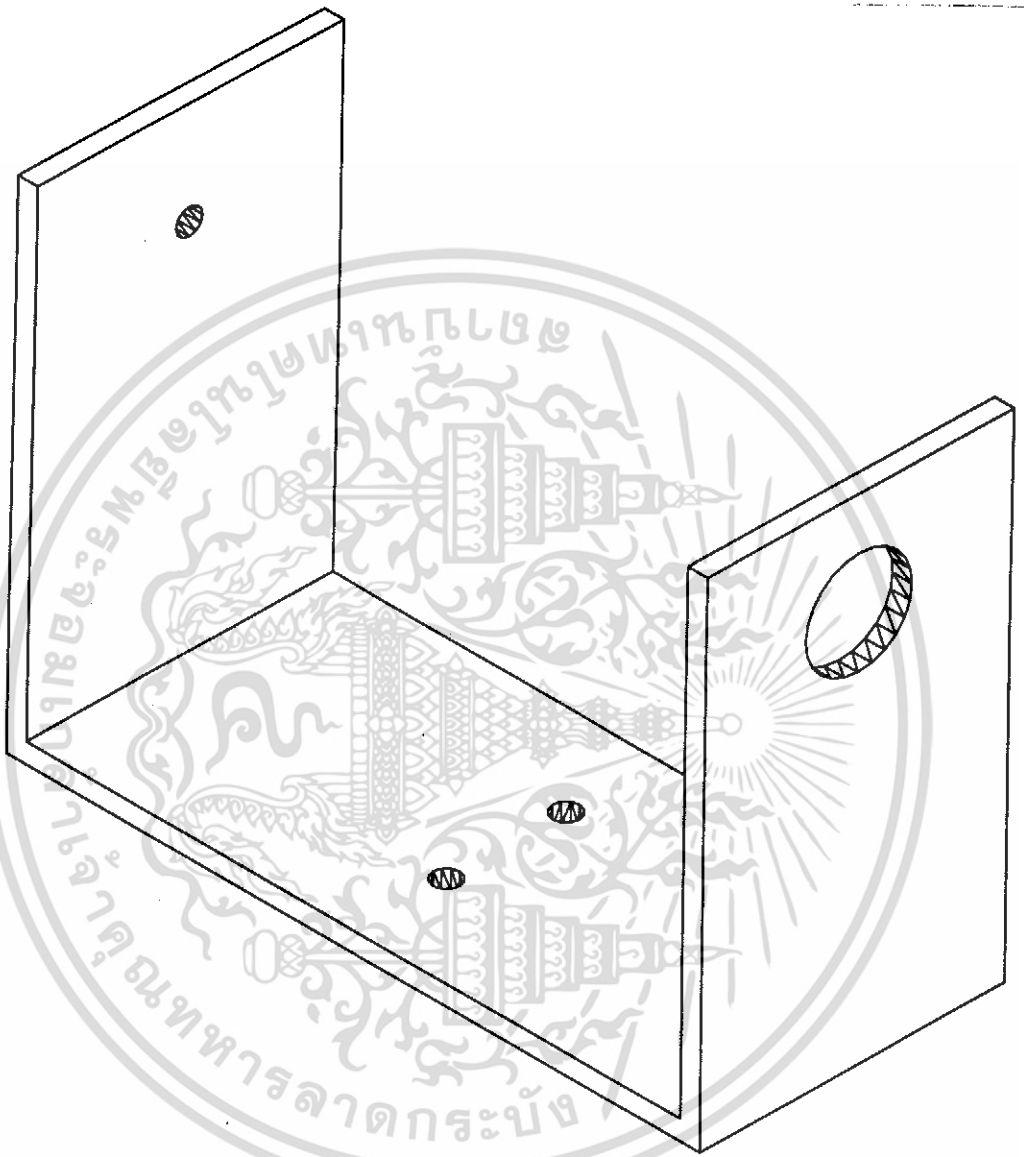
SIDE VIEW



TOP VIEW

NO. of Piece	Nomenclature : Module	Scale 1 : 1
4/11	List of Parts : Part2	Top View/ Front View/ Side View
	King Mongkut's Institute of Tecnology	Date : 16/10/2005

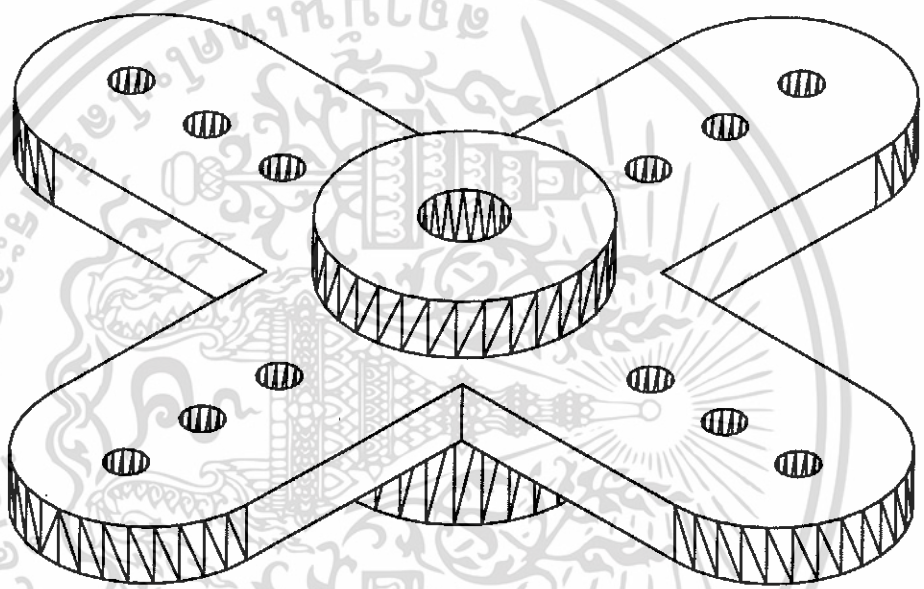
A4



NO. of Piece	Nomenclature : Module	Scale 4 : 1
5/11	List of Parts : Part2	3D View
King Mongkut's Institute of Tecnology		Date : 16/10/2005

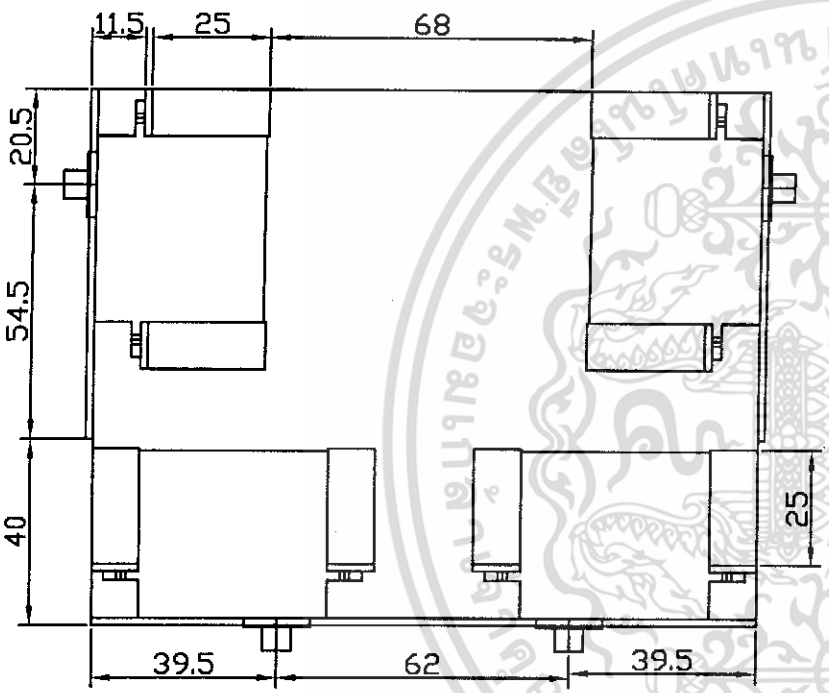
ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ห้ามนำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

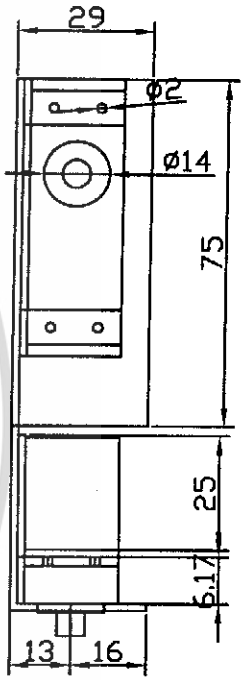


NO. of Piece	Nomenclature : Module	Scale 10 : 1
6/11	List of Parts : Part3	3D View
King Mongkut's Institute of Tecnology		Date : 16/10/2005

M16

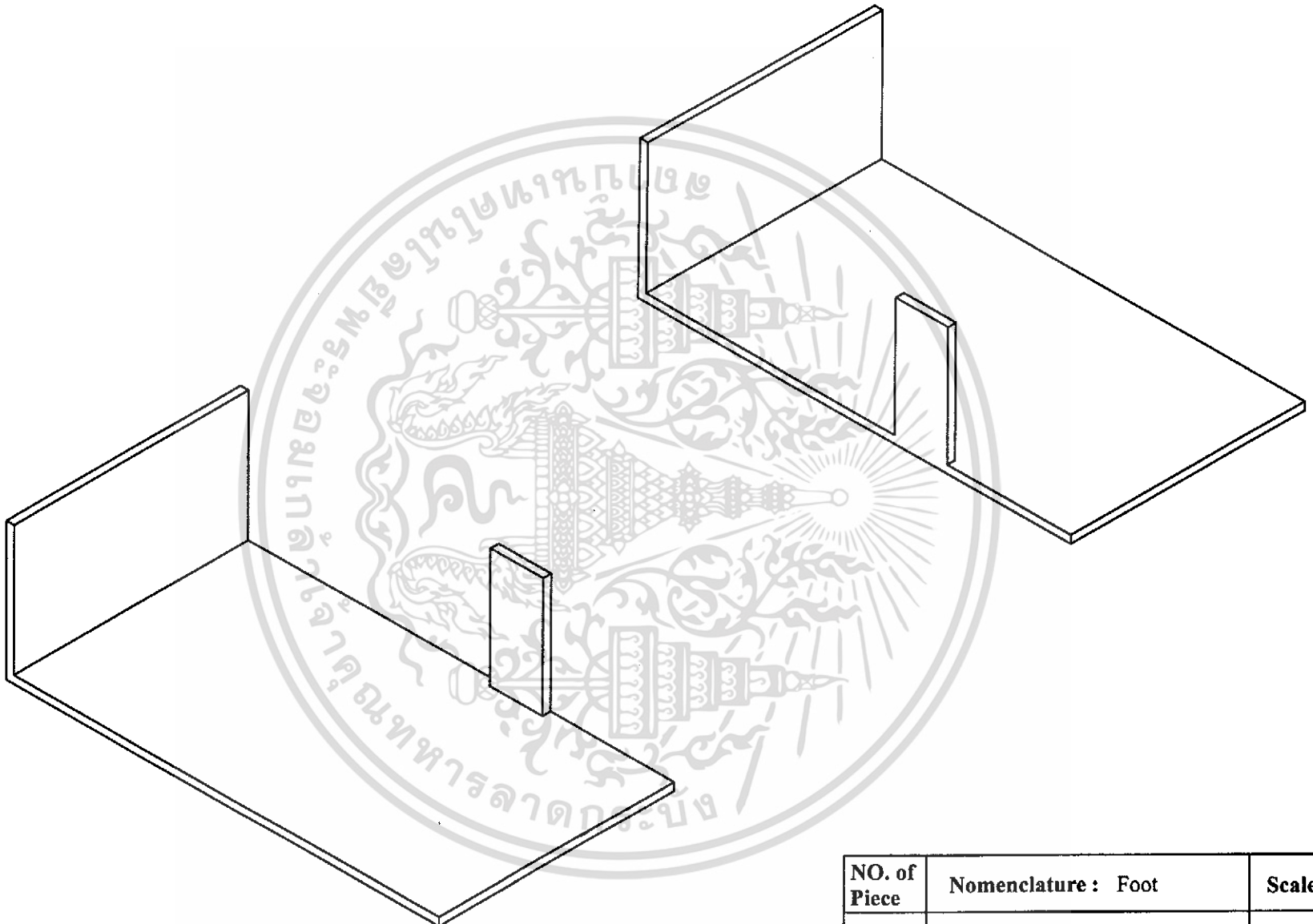


FRONT VIEW

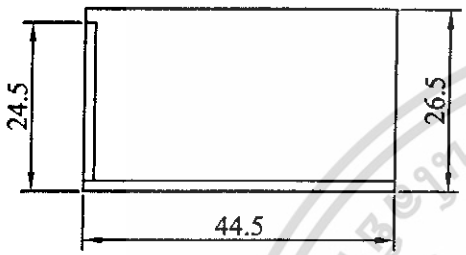


SIDE VIEW

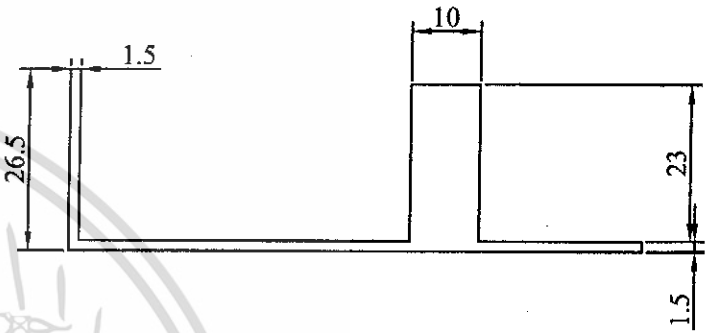
NO. of Piece	Nomenclature : Body	Scale 1 : 1
7/11	List of Parts : Body	Front View/Side View
King Mongkut's Institute of Tecnology		Date : 16/10/2005



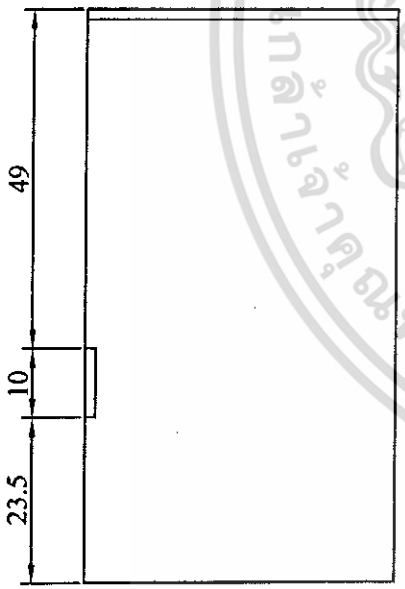
NO. of Piece	Nomenclature : Foot	Scale 2 : 1
8/11	List of Parts : Foot	3D View
King Mongkut's Institute of Tecnology		Date : 16/10/2005



FRONT VIEW



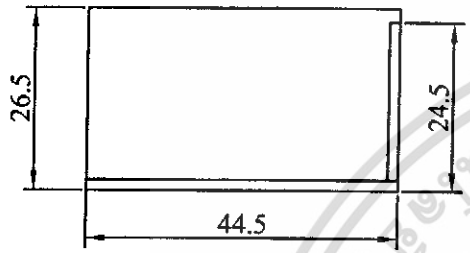
SIDE VIEW



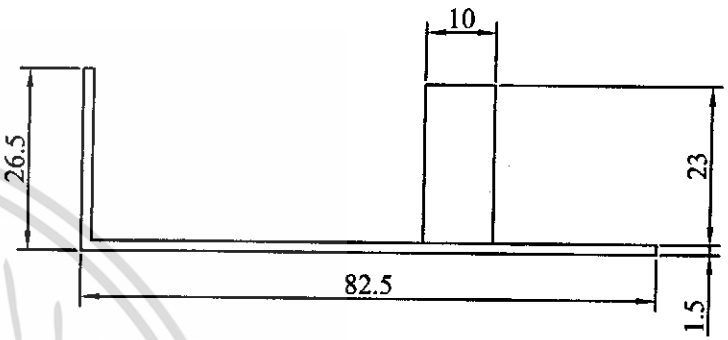
TOP VIEW

NO. of Piece	Nomenclature : Foot	Scale 1 : 1
9/11	List of Parts : Left Foot	FrontView/Top View/ Side View
King Mongkut's Institute of Tecnology		Date : 16/10/2005

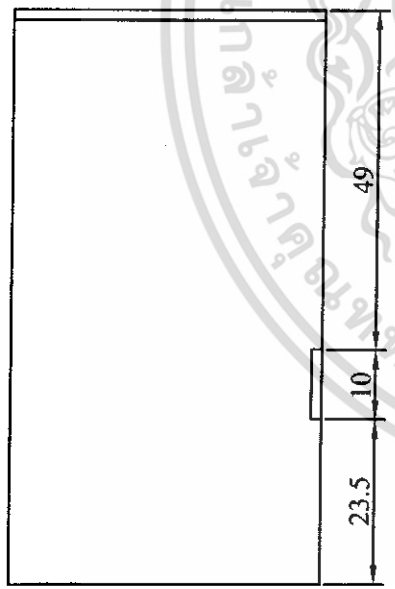
MM 9



FRONT VIEW



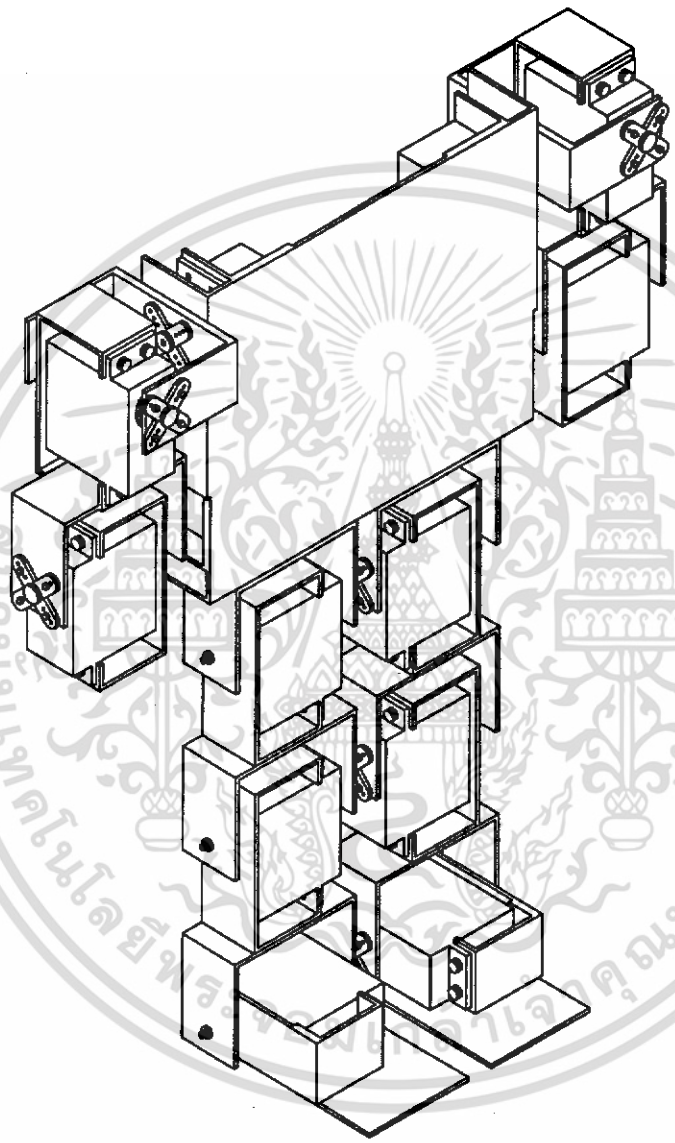
SIDE VIEW



TOP VIEW

NO. of Piece	Nomenclature : Foot	Scale 1 : 1
10/11	List of Parts : Right Foot	FrontView/Top View/ Side View
King Mongkut's Institute of Tecnology		Date : 16/10/2005

หน้า 10

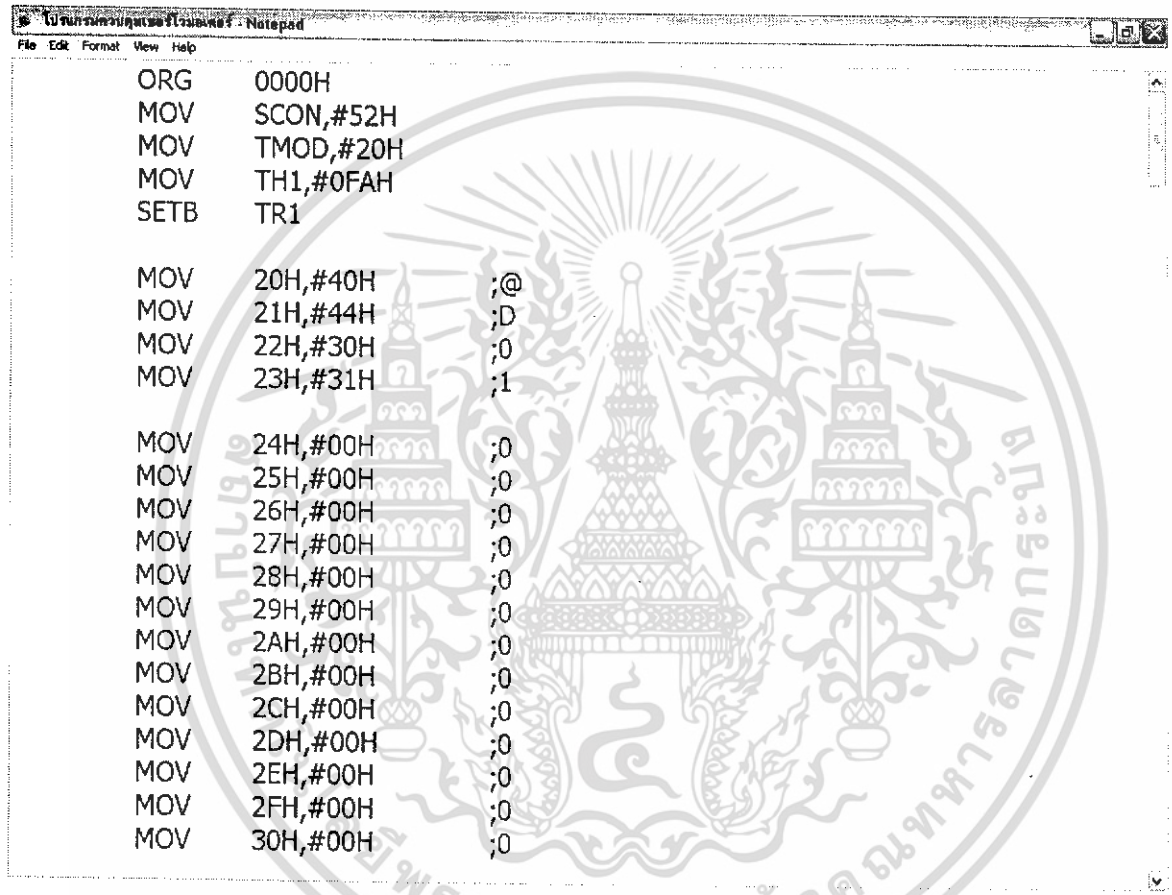


NO. of Piece	Nomenclature : Robot	Scale 1 : 1
11/11	List of Parts : Robot	3D View
	King Mongkut's Institute of Tecnology	Date : 16/10/2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ซอร์ทโค้ดโปรแกรมภาษาแอสแซมบลีที่ใช้ควบคุมเซอร์โวมอเตอร์



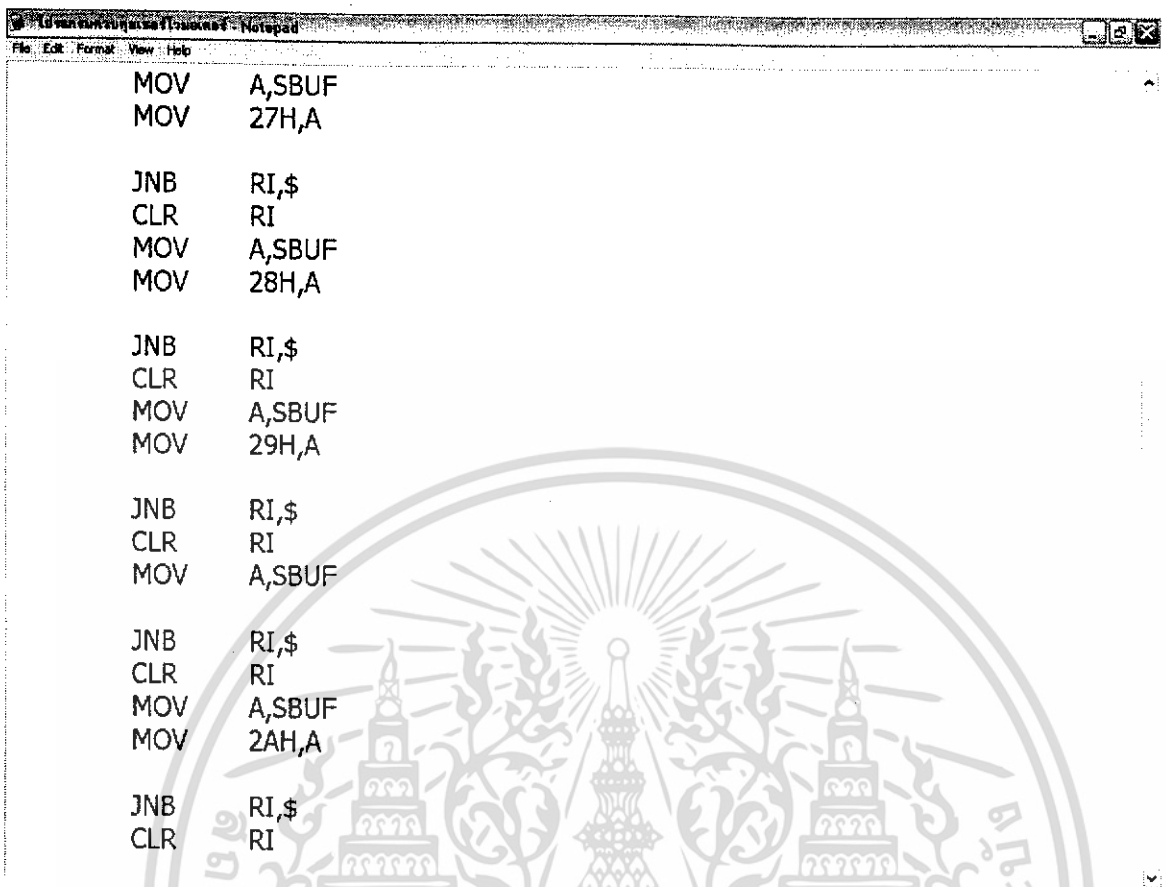
```
ORG     0000H
MOV     SCON,#52H
MOV     TMOD,#20H
MOV     TH1,#0FAH
SETB    TR1

MOV     20H,#40H        ;@
MOV     21H,#44H        ;D
MOV     22H,#30H        ;0
MOV     23H,#31H        ;1

MOV     24H,#00H        ;0
MOV     25H,#00H        ;0
MOV     26H,#00H        ;0
MOV     27H,#00H        ;0
MOV     28H,#00H        ;0
MOV     29H,#00H        ;0
MOV     2AH,#00H        ;0
MOV     2BH,#00H        ;0
MOV     2CH,#00H        ;0
MOV     2DH,#00H        ;0
MOV     2EH,#00H        ;0
MOV     2FH,#00H        ;0
MOV     30H,#00H        ;0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

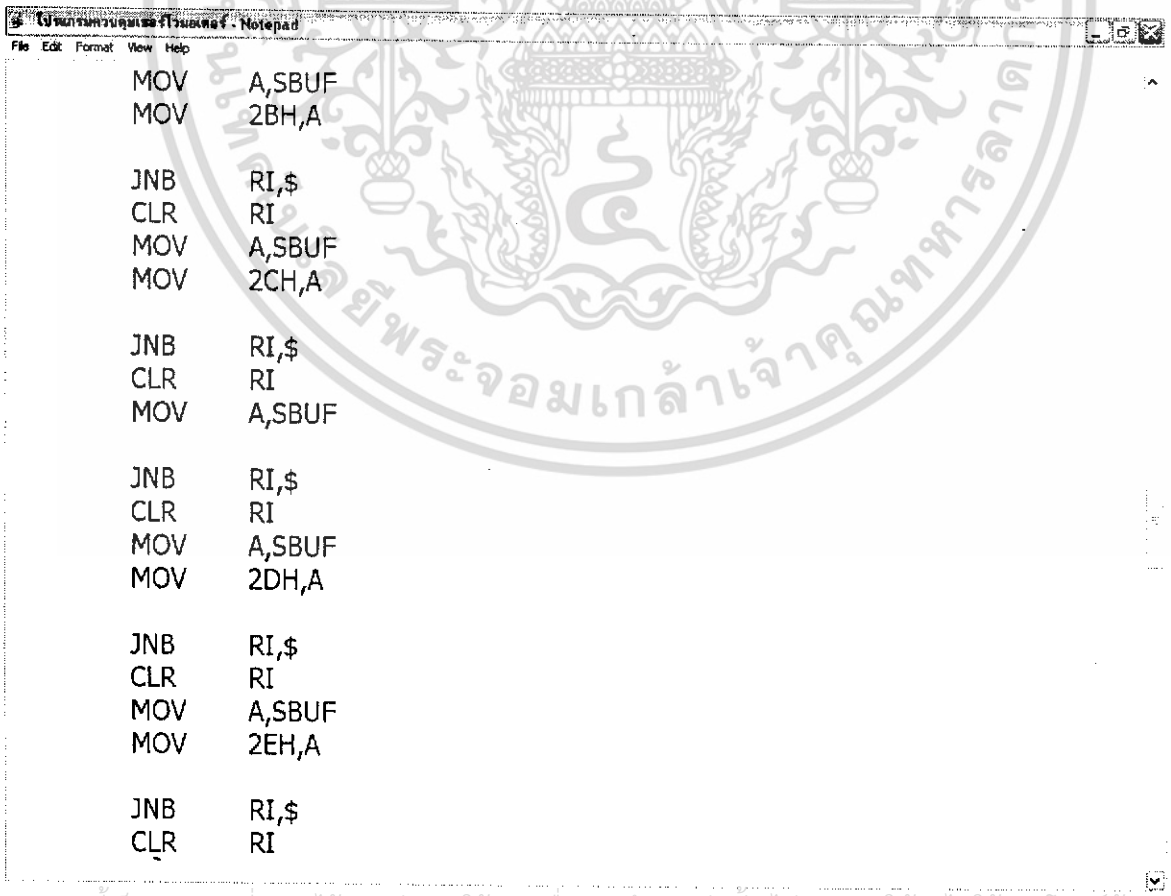
```
        MOV     31H,#00H        ;0

STO:    CALL    SERVO
ST:     JNB     RI,$
        CLR     RI
        MOV     A,SBUF
        XRL     A,20H
        JNZ     ST

        JNB     RI,$
        CLR     RI
        MOV     A,SBUF
        XRL     A,21H
        JNZ     ST

        JNB     RI,$
        CLR     RI
        MOV     A,SBUF
        XRL     A,22H
        JNZ     ST

        JNB     RI,$
        CLR     RI
        MOV     A,SBUF
```

```
        XRL     A,23H
        JNZ     ST

        JNB     RI,$
        CLR     RI
        MOV     A,SBUF
        MOV     24H,A

        JNB     RI,$
        CLR     RI
        MOV     A,SBUF
        MOV     25H,A

        JNB     RI,$
        CLR     RI
        MOV     A,SBUF
        MOV     26H,A

        JNB     RI,$
        CLR     RI
        MOV     A,SBUF

        JNB     RI,$
        CLR     RI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

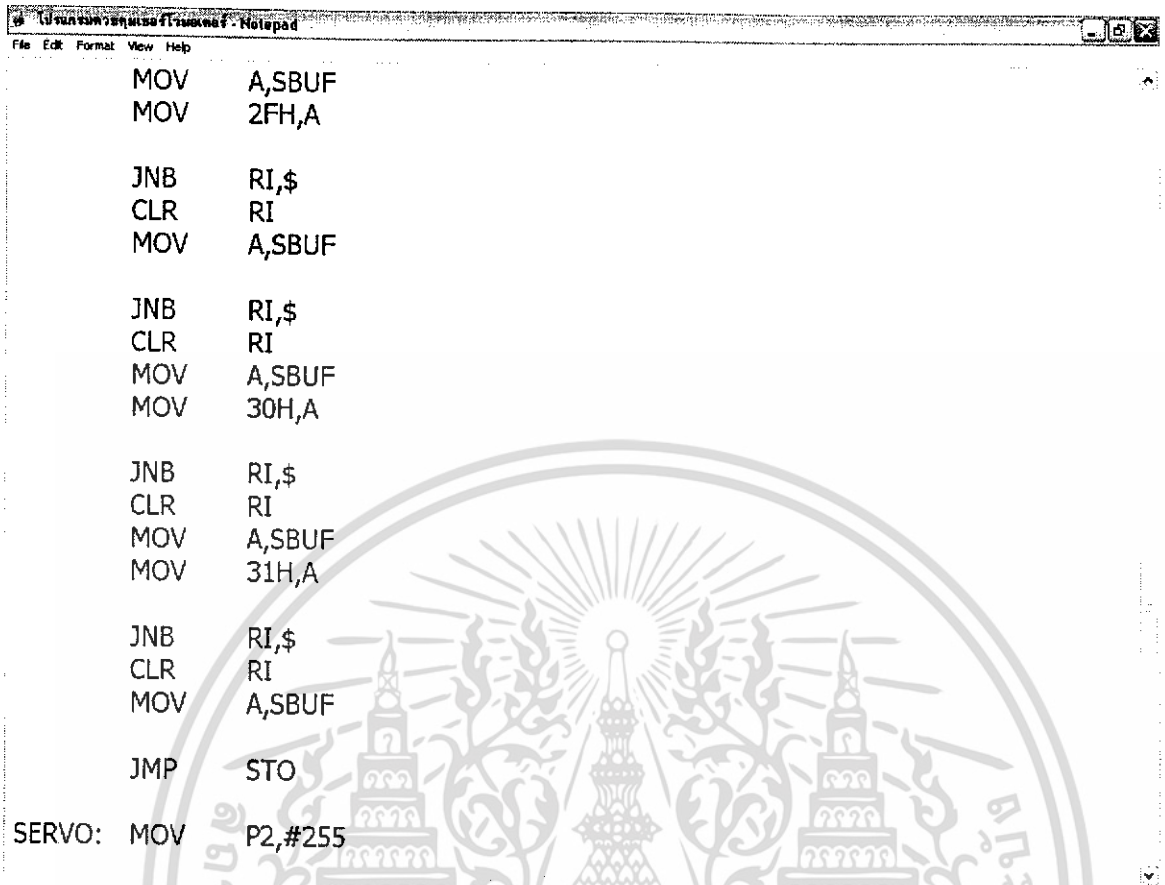


```
MOV     A,SBUF
MOV     27H,A

JNB     RI,$
CLR     RI
MOV     A,SBUF
MOV     28H,A

JNB     RI,$
CLR     RI
MOV     A,SBUF
MOV     29H,A

JNB     RI,$
CLR     RI
MOV     A,SBUF

JNB     RI,$
CLR     RI
MOV     A,SBUF
MOV     2AH,A

JNB     RI,$
CLR     RI
```

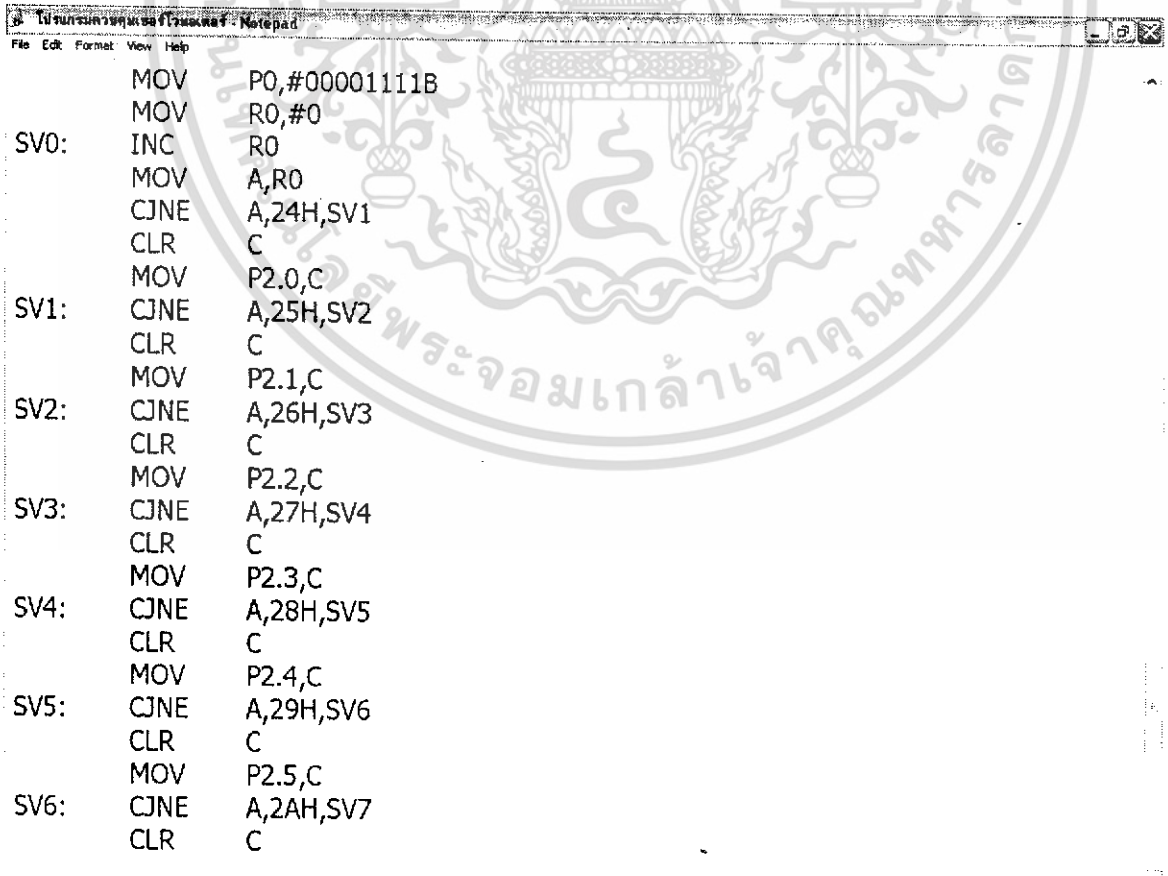


```
MOV     A,SBUF
MOV     2BH,A

JNB     RI,$
CLR     RI
MOV     A,SBUF
MOV     2CH,A

JNB     RI,$
CLR     RI
MOV     A,SBUF

JNB     RI,$
CLR     RI
MOV     A,SBUF
MOV     2DH,A

JNB     RI,$
CLR     RI
MOV     A,SBUF
MOV     2EH,A

JNB     RI,$
CLR     RI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

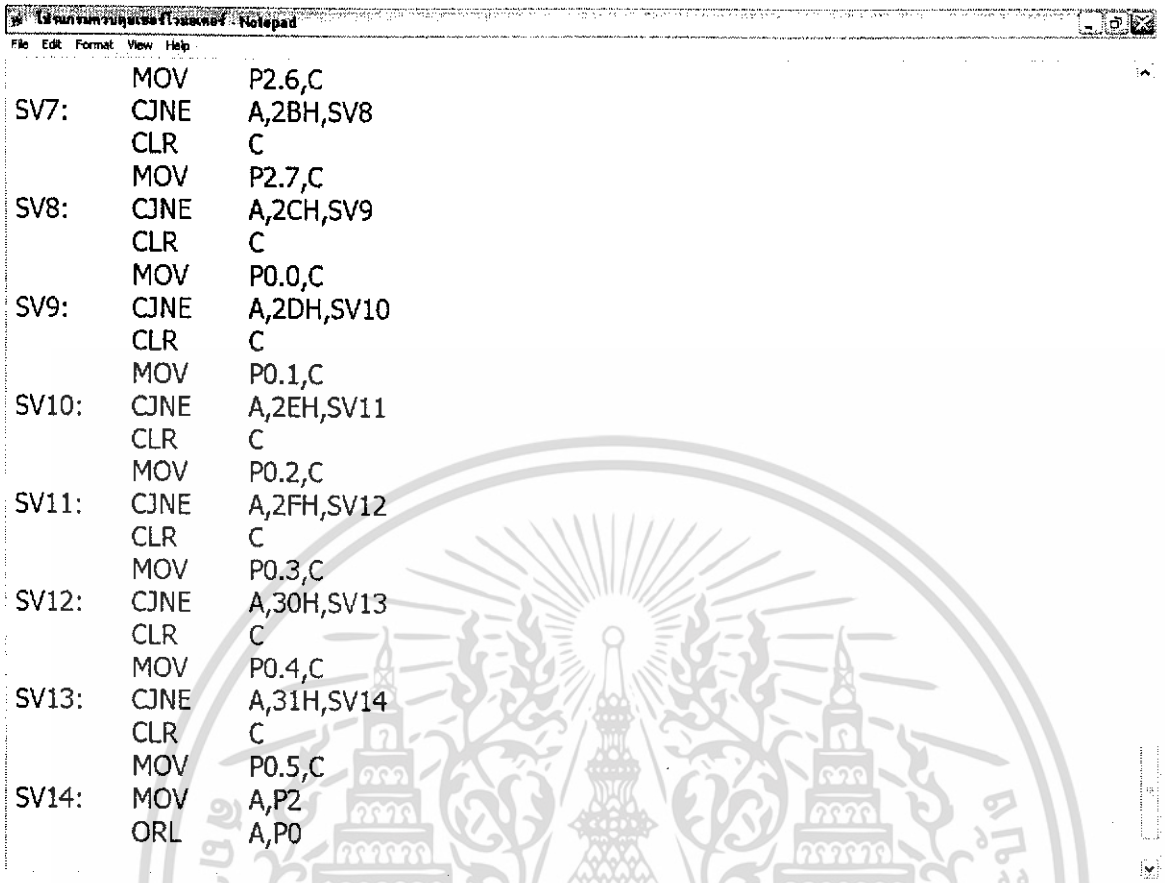


```
File Edit Format View Help
        MOV     A,SBUF
        MOV     2FH,A

        JNB     RI,$
        CLR     RI
        MOV     A,SBUF

        JNB     RI,$
        CLR     RI
        MOV     A,SBUF
        MOV     30H,A

        JNB     RI,$
        CLR     RI
        MOV     A,SBUF
        MOV     31H,A

        JNB     RI,$
        CLR     RI
        MOV     A,SBUF

        JMP     STO

SERVO:  MOV     P2,#255
```



```
File Edit Format View Help
        MOV     P0,#00001111B
        MOV     R0,#0
SV0:    INC     R0
        MOV     A,R0
        CJNE    A,24H,SV1
        CLR     C
        MOV     P2.0,C
SV1:    CJNE    A,25H,SV2
        CLR     C
        MOV     P2.1,C
SV2:    CJNE    A,26H,SV3
        CLR     C
        MOV     P2.2,C
SV3:    CJNE    A,27H,SV4
        CLR     C
        MOV     P2.3,C
SV4:    CJNE    A,28H,SV5
        CLR     C
        MOV     P2.4,C
SV5:    CJNE    A,29H,SV6
        CLR     C
        MOV     P2.5,C
SV6:    CJNE    A,2AH,SV7
        CLR     C
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
        MOV     P2.6,C
SV7:    CJNE    A,2BH,SV8
        CLR     C
        MOV     P2.7,C
SV8:    CJNE    A,2CH,SV9
        CLR     C
        MOV     P0.0,C
SV9:    CJNE    A,2DH,SV10
        CLR     C
        MOV     P0.1,C
SV10:   CJNE    A,2EH,SV11
        CLR     C
        MOV     P0.2,C
SV11:   CJNE    A,2FH,SV12
        CLR     C
        MOV     P0.3,C
SV12:   CJNE    A,30H,SV13
        CLR     C
        MOV     P0.4,C
SV13:   CJNE    A,31H,SV14
        CLR     C
        MOV     P0.5,C
SV14:   MOV     A,P2
        ORL     A,P0
```

```
        JNZ     SV0
        RET

        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

atures

- ompatible with MCS-51™ Products
- Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- lly Static Operation: 0 Hz to 24 MHz
- ree-Level Program Memory Lock
- 8 x 8-Bit Internal RAM
- Programmable I/O Lines
- o 16-Bit Timer/Counters
- x Interrupt Sources
- ogrammable Serial Channel
- w Power Idle and Power Down Modes

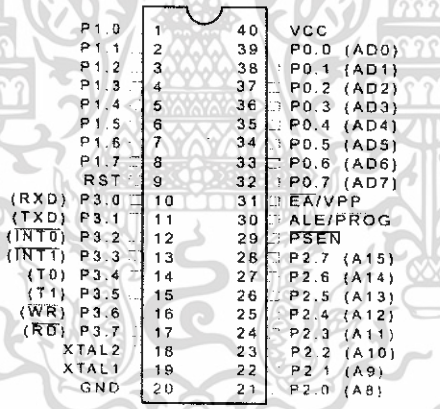
scription

AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K s of Flash Programmable and Erasable Read Only Memory (PEROM). The ce is manufactured using Atmel's high density nonvolatile memory technology is compatible with the industry standard MCS-51™ instruction set and pinout. The hip Flash allows the program memory to be reprogrammed in-system or by a con- ional nonvolatile memory programmer. By combining a versatile 8-bit CPU with h on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which ides a highly flexible and cost effective solution to many embedded control appli- ns.

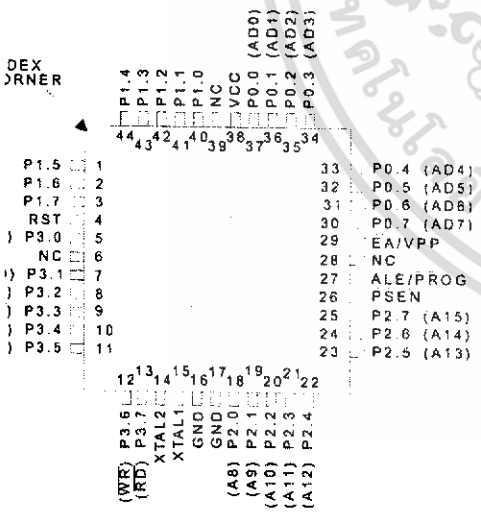
(continued)

Configurations

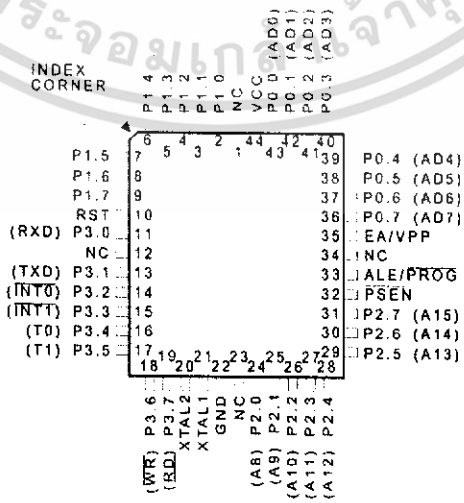
PDIP



PQFP/TQFP



PLCC



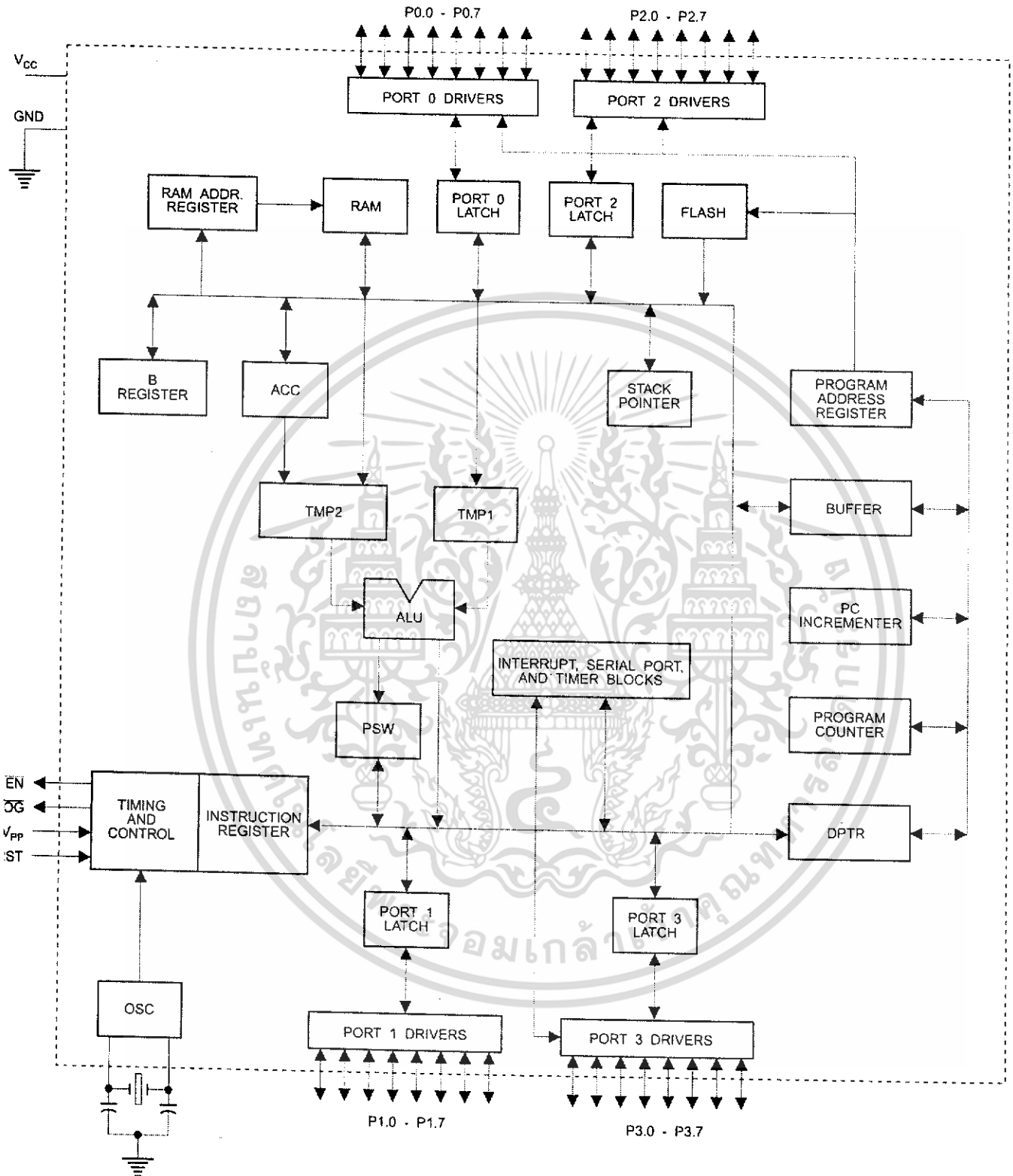
8-Bit Microcontroller with 4K Bytes Flash

AT89C51

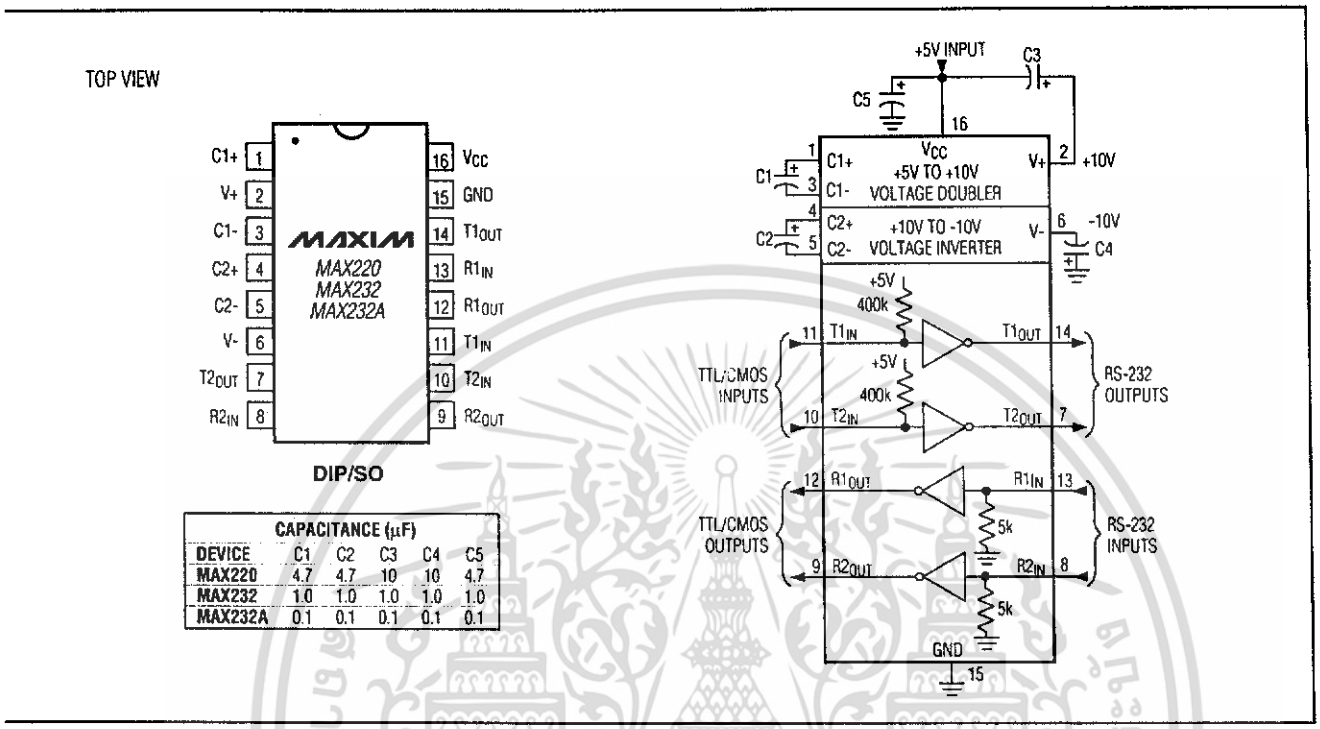
0265F-A–12/97

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

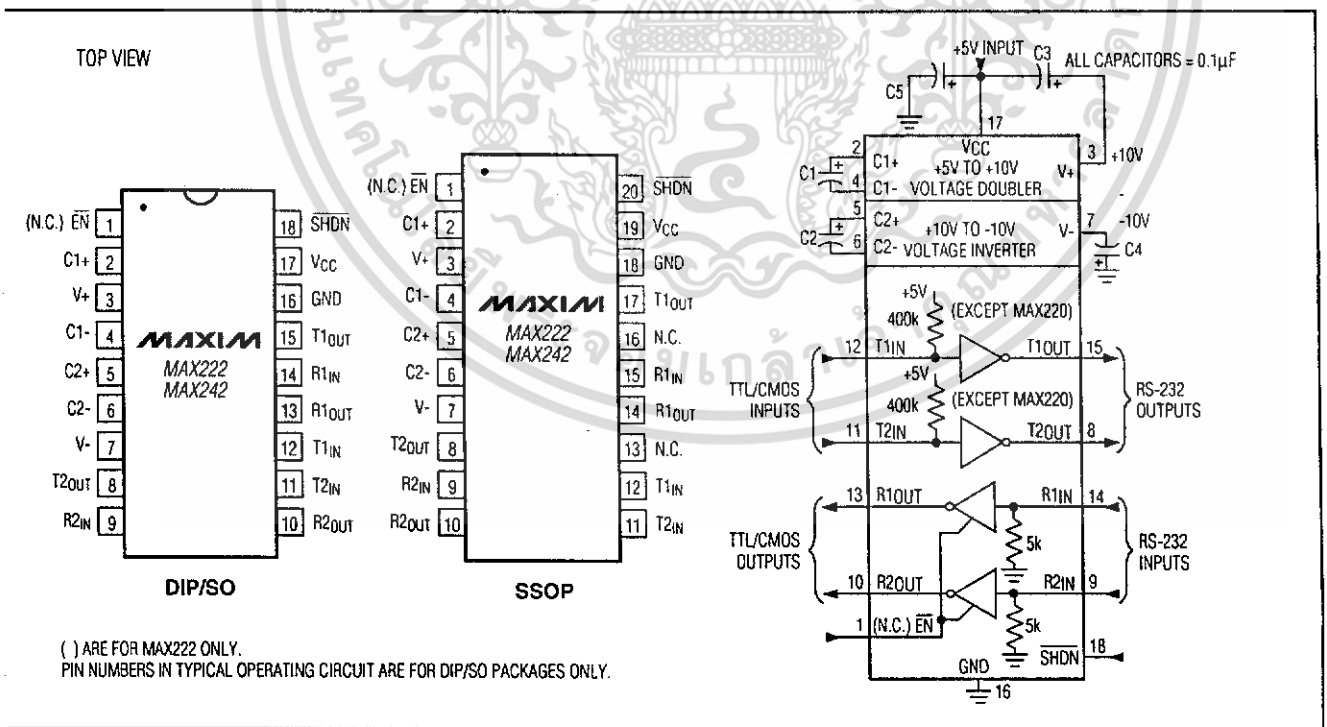
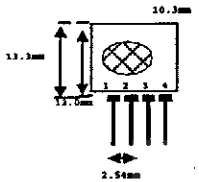


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Easy-Link Wireless



pin 1 : GND
pin 2 : Data In
pin 3 : Vcc
pin 4 : Antenna (RF output)

Frequency 315, 418 and 433.92 Mhz

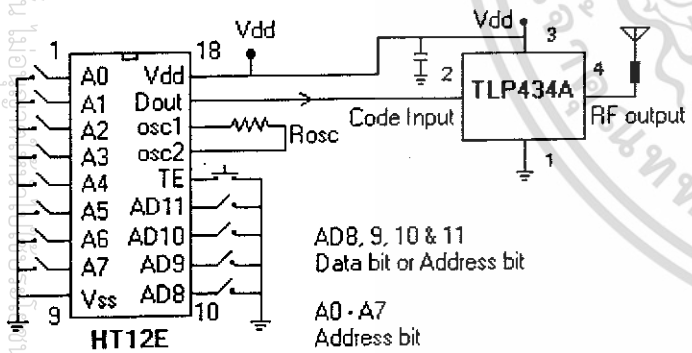
Modulation : ASK
Operation Voltage : 2 - 12 VDC

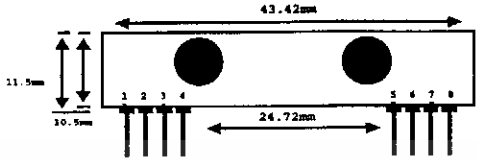
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Vcc	Operating supply voltage		2.0	-	12.0	V
Icc 1	Peak Current (2V)		-	-	1.64	mA
Icc 2	Peak Current (12V)		-	-	19.4	mA
Vh	Input High Voltage	Idata= 100uA (High)	Vcc-0.5	Vcc	Vcc+0.5	V
Vl	Input Low Voltage	Idata= 0 uA (Low)	-	-	0.3	V
FO	Absolute Frequency	315Mhz module	314.8	315	315.2	MHz
PO	RF Output Power- 50ohm	Vcc = 9V-12V	-	16	-	dBm
		Vcc = 5V-6V	-	14	-	dBm
DR	Data Rate	External Encoding	512	4.8K	200K	bps

Notes : (Case Temperature = 25°C +- 2°C , Test Load Impedance = 50 ohm)

Application Circuit :

Typical Key-chain Transmitter using HT12E-18DIP, a Binary 12 bit Encoder from Holtek Semiconductor Inc.





pin 1 : Gnd
pin 2 : Digital Data Output
pin 3 : Linear Output /Test
pin 4 : Vcc
pin 5 : Vcc
pin 6 : Gnd
pin 7 : Gnd
pin 8 : Antenna

Frequency 315, 418 and 433.92 Mhz

Modulation : ASK
Supply Voltage : 3.3 - 6.0 VDC
Output : Digital & Linear

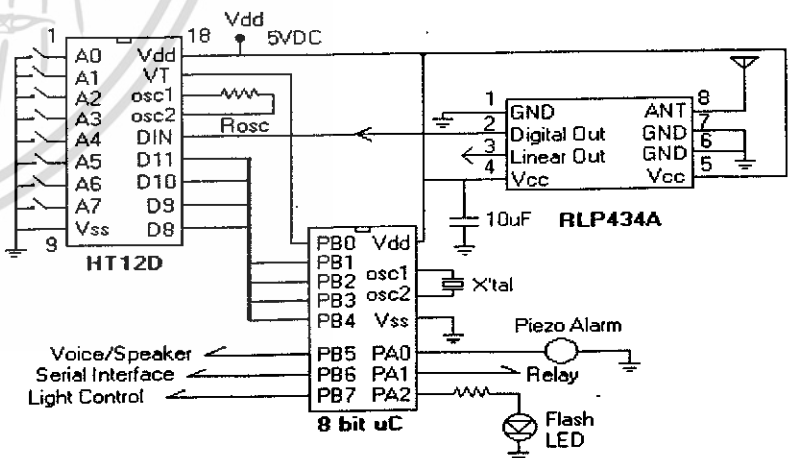
Symbol	Parameter	Conditions	Min	Typ	Max	
Vcc	Operating supply voltage		3.3	5.0V	6.0	V
Itot	Operating Current		-	4.5		mA
Vdata	Data Out	Idata = +200 uA (High)	Vcc-0.5	-	Vcc	V
		Idata = -10 uA (Low)	-	-	0.3	V

Electrical Characteristics

Characteristics	SYM	Min	Typ	Max	Unit
Operation Radio Frequency	FC		315, 418 and 433.92		MHz
Sensitivity	Pref		-110		dBm
Channel Width			+-500		Khz
Noise Equivalent BW			4		Khz
Receiver Turn On Time			5		ms
Operation Temperature	Top	-20	-	80	C
Baseboard Data Rate			4.8		KHz

Application Circuit :

Typical RF Receiver using HT12D-18DIP, a Binary 12 bit Decoder with 8 bit uC HT48RXX from Holtek Semiconductor Inc.



Laipac Technology, Inc.
105 West Beaver Creek Rd. Unit 207 Richmond Hill Ontario L4B 1C6 Canada
Tel: (905)762-1228 Fax: (905)763-1737 e-mail: info@laipac.com

LAIPAC TECH